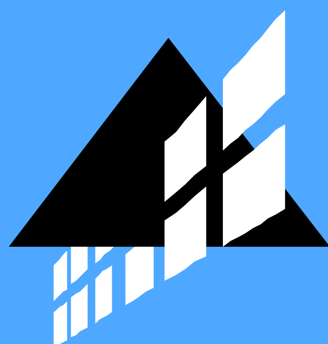
Manual de Utilização do MasterTool IEC MT8200

Rev. A 01/2009

Cód. Doc.: MU299606



altus

Conforme o Código de Defesa do Consumidor vigente no Brasil, informamos a seguir, aos clientes que utilizam nossos produtos, aspectos relacionados com a segurança de pessoas e instalações.

Os equipamentos de automação industrial fabricados pela Altus são robustos e confiáveis devido ao rígido controle de qualidade a que são submetidos. No entanto, equipamentos eletrônicos de controle industrial (controladores programáveis, comandos numéricos, etc.) podem causar danos às máquinas ou processos por eles controlados em caso de defeito em suas partes e peças ou de erros de programação ou instalação, podendo inclusive colocar em risco vidas humanas.

O usuário deve analisar as possíveis conseqüências destes defeitos e providenciar instalações adicionais externas de segurança que, em caso de necessidade, sirvam para preservar a segurança do sistema, principalmente nos casos da instalação inicial e de testes.

Os equipamentos fabricados pela Altus não trazem riscos ambientais diretos, não emitindo nenhum tipo de poluente durante sua utilização. No entanto, no que se refere ao descarte dos equipamentos, é importante salientar que quaisquer componentes eletrônicos incorporados em produtos contêm materiais nocivos à natureza quando descartados de forma inadequada. Recomenda-se, portanto, que quando da inutilização deste tipo de produto, o mesmo seja encaminhado para usinas de reciclagem que dêem o devido tratamento para os resíduos.

É imprescindível a leitura completa dos manuais e/ou características técnicas do produto antes da instalação ou utilização do mesmo.

A Altus garante os seus equipamentos conforme descrito nas Condições Gerais de Fornecimento, anexada às propostas comerciais.

A Altus garante que seus equipamentos funcionam de acordo com as descrições contidas explicitamente nos manuais e/ou características técnicas, não garantindo a satisfação de algum tipo particular de aplicação dos equipamentos.

A Altus desconsiderará qualquer outra garantia direta ou implícita, principalmente quando se tratar de fornecimento de terceiros.

Pedidos de informações adicionais sobre o fornecimento e/ou características dos equipamentos e serviços Altus devem ser feitos por escrito. A Altus não se responsabiliza por informações fornecidas sobre seus equipamentos sem registro formal.

DIREITOS AUTORAIS

Sumário

1. Introdução

O software MasterTool IEC é um ambiente de desenvolvimento completo para controladores programáveis utilizado na Série Duo. Esta ferramenta permite o monitoramento de processos, configuração de módulos e simulação de aplicações. É executável nos ambientes Windows 2000® , Windows XP®, Windows Vista® tendo disponíveis versões em inglês, espanhol e português do software. A programação da aplicação é feita conforme a norma IEC 61131, tendo disponível para isso as 5 linguagens de programação de controladores programáveis previstas nessa. A edição da aplicação é feita integralmente na ferramenta sendo possível utilizar mais de uma linguagem para a mesma aplicação. No software é possível converter de uma linguagem para outra através do editor.

No MasterTool IEC, a programação é realizada utilizando o conceito de programação simbólica, englobando estruturas de dados e criação de novos tipos dados baseados nos já existentes. A configuração do hardware do controlador programável é feita através de uma árvore intuitiva e de fácil utilização. Possibilita a monitoração das variáveis em execução e, além disso, possui simulador da aplicação, o que facilita e agiliza o desenvolvimento, permitindo a depuração de lógicas, além de possibilitar testes sem a necessidade de equipamento. Com a opção de carregar os arquivos fonte no equipamento, é possível fazer comentários na aplicação com descrições das lógicas implementadas e se for necessário pode ser realizada a leitura do que foi carrgado nele. O MasterTool IEC permite a programação da IHM incorporada ao controlador, sem a necessidade de uma outra ferramenta.

Documentos Relacionados a este Manual

Para obter informações adicionais sobre o MasterTool IEC podem ser consultados outros documentos específicos além deste. Estes documentos encontram-se disponíveis na sua última revisão em www.altus.com.br.

Considerações gerais sobre documentação ALTUS

Cada produto possui um documento denominado Característica Técnica (CT), onde se encontram as características do produto em questão. Adicionalmente o produto pode possuir Manuais de Utilização (os códigos dos manuais são citados na CT).

Por exemplo, o módulo PO2022 tem todas as informações de características de utilização e de compra, na sua CT. Já o PO5063 possui, além da CT, um manual de utilização.

Documentação de suporte ao MasterTool IEC:

Aconselham-se os seguintes documentos como fontes de informação adicional:

Manual de Utilização do MasterTool IEC (MU299606)	Manual de utilização do programador MasterTool IEC (este documento)
Manual de Utilização da Visualização no MasterTool IEC (MU299608)	Manual de utilização do sistema de visualização do programador MasterTool IEC, incluindo MasterTool IEC HMI e Visualização Dispositivo/Web.
Manual de Utilização DU350/DU351	Manual de utilização dos controladores com IHM integrada DU350 e DU351 que utilizam como programador o MasterTool IEC.

Tabela 1-1. Documentação de suporte

Inspeção Visual

Antes de proceder à instalação, é recomendável fazer uma inspeção visual cuidadosa dos equipamentos, verificando se não há danos causados pelo transporte. Verifique se todos os

componentes de seu pedido estão em perfeito estado. Em caso de defeitos, informe a companhia transportadora e o representante ou distribuidor Altus mais próximo.

CUIDADO:
Antes de retirar os módulos da embalagem, é importante descarregar eventuais potenciais estáticos acumulados no corpo. Para isso, toque (com as mãos nuas) em qualquer superfície metálica aterrada antes de manipular os módulos. Tal procedimento garante que os níveis de eletricidade estática suportados pelo módulo não serão ultrapassados.

É importante registrar o número de série de cada equipamento recebido, bem como as revisões de software, caso existentes. Essas informações serão necessárias caso seja preciso contatar o Suporte Técnico da Altus.

Suporte Técnico

Para entrar em contato com o Suporte Técnico da Altus em São Leopoldo, RS, ligue para +55-51-3589-9500. Para conhecer os centros de Suporte Técnico da Altus existentes em outras localidades, consulte nosso site (www.altus.com.br) ou envie um email para altus@altus.com.br.

Se o equipamento já estiver instalado tenha em mãos as seguintes informações ao solicitar assistência:

- os modelos dos equipamentos utilizados e a configuração do sistema instalado;
- o número de série da UCP;
- a revisão do equipamento e a versão do software executivo, constantes na etiqueta afixada na lateral do produto;
- as informações sobre o modo de operação da UCP, obtidas através do programador MasterTool;
- o conteúdo do programa aplicativo (módulos), obtido através do programador MasterTool;
- a versão do programador utilizado.

Mensagens de Advertência Utilizadas neste Manual

Neste manual, as mensagens de advertência apresentarão os seguintes formatos e significados:

PERIGO:
Relatam causas potenciais, que se não observadas, *levam* a danos à integridade física e saúde, patrimônio, meio ambiente e perda da produção.

CUIDADO:
Relatam detalhes de configuração, aplicação e instalação que *devem* ser seguidos para evitar condições que possam levar a falha do sistema e suas conseqüências relacionadas.

ATENÇÃO:
Indicam detalhes importantes de configuração, aplicação ou instalação para obtenção da máxima performance operacional do sistema.

Visão geral das funções do MasterTool IEC

Como um projeto é estruturado

Um projeto é associado a um arquivo apropriado. A primeira POU (Unidade de Organização de Programa) criada em um novo projeto será automaticamente nomeada PLC_PRG. O processo inicia aqui (em conformidade com a função principal em um programa C) e outras POUs podem ser acessadas a partir do mesmo ponto (programas, blocos funcionais e funções).

Uma vez definida a Configuração da Tarefa, não é necessário criar um programa nomeado PLC_PRG. Mais informações sobre isto estão disponíveis no capítulo Configuração de tarefas.

Há diferentes tipos de objetos em um projeto: POUs, tipo de dados, elementos de display (visualizações) e recursos.

O Organizador de objetos contém a lista de todos os objetos no seu projeto.

Como configurar meu projeto?

Primeiro deve-se configurar o CP para verificar a exatidão dos endereços usados no projeto.

Então, pode-se criar as POUs necessárias para resolver o problema.

Agora pode-se programar as POUs necessárias nas linguagens desejadas.

Quando a programação estiver concluída, pode-se compilar o projeto e remover erros, se houver.

Como testar meu projeto?

Depois que todos os erros foram removidos, deve-se ativar a **simulação** e fazer login no CP simulado (modo Online).

Em seguida deve-se abrir a janela com a configuração do CP e testar o projeto para a seqüência correta. Para fazer isso, deve-se introduzir variáveis de entrada "manualmente" e observar se as saídas estão conforme o esperado. Também é possível observar a seqüência de valores das variáveis locais nas POUs. No **Gerenciador de Monitoração e Receitas**, pode-se configurar dados de registro cujos valores se deseja analisar.

Depuração

Em caso de um erro de programação, podem ser definidos breakpoints. Se o processo pára em um breakpoint, nesse momento pode-se examinar os valores de todas as variáveis de projeto. Trabalhando seqüencialmente (passo a passo) pode-se verificar a exatidão lógica do programa.

Funções online adicionais

Pode ser utilizado o **controle de fluxo** para verificar quais linhas de programa foram executadas.

Se ativado nas configurações do dispositivo, o **Rastreamento de Amostragem** permite rastrear e exibir o comportamento atual das variáveis durante um período estendido de tempo.

Também uma função específica do dispositivo é o **Navegador CP**, que pode servir para solicitar certas informações do CP.

Uma vez que o projeto foi configurado e testado, ele pode ser carregado para o hardware e testado também. As mesmas funções online usadas no modo de simulação estarão disponíveis.

Características adicionais do MasterTool IEC

O projeto inteiro pode ser documentado ou exportado para um arquivo de texto em qualquer momento.

Usando as **configurações do dispositivo** adequadas, que podem ser carregadas com a ajuda de um arquivo do dispositivo (pacote de suporte do dispositivo), pode-se carregar o mesmo projeto do MasterTool IEC para vários sistemas do dispositivo.

Variáveis globais de rede e um **Gerenciador de Parâmetros** podem estar disponíveis, se ativado pelas configurações atuais do dispositivo, para troca de dados em uma rede de controladores.

Ferramentas: esta funcionalidade também é dependente do dispositivo e permite iniciar seus arquivos executáveis específicos em um projeto MasterTool IEC. Além disso, podem ser definidos os arquivos que seram carregados para o controlador. Conexões a ferramentas externas podem ser predefinidas no arquivo do dispositivo e/ou incluídas na árvore de Recursos de projeto.

Uma **tela do MasterTool IEC** pode ser processada especificamente para o dispositivo afim de disponibilizá-la como **tela do Dispositivo**. Isso permite executar e exibir a tela através de um visor no CP.

2. Conceitos associados ao MasterTool IEC

Componentes de projeto

Projeto

Um projeto contém todos os objetos em um programa do CP. Ele é salvo em um arquivo nomeado depois do projeto. Os seguintes objetos estão incluídos nele: POUs (Unidades de Organização de Programa), tipo de dados, telas, recursos e bibliotecas.

POU (Unidade de Organização de Programa)

Funções, blocos funcionais e programas são POUs que podem ser complementadas por ações.

Cada POU consiste de uma parte de declaração e de um corpo. O corpo é gerado em uma das linguagens de programação IEC que incluem IL, ST, SFC, FBD, LD ou CFC.

O **MasterTool IEC** suporta todas as POUs padronizadas pela IEC. Para usa-las no projeto, deve-se incluir a biblioteca standard.lib.

POUs podem chamar outras POUs. Entretanto, recursividade não é permitida.

Função

Uma ‘função’ é uma POU que fornece exatamente um elemento de dados (que pode consistir de vários elementos, como campos ou estruturas) quando esta é processada. A chamada da função na linguagem textual pode ocorrer como um operador em expressões.

Ao declarar uma função, ela deve receber um tipo. Isto significa que depois do nome da função, deve-se digitar dois pontos (:), seguidos de um tipo.

Uma declaração de função correta se parece com o exemplo a seguir:

```
FUNCTION Fct: INT
```

Além disso, um resultado deve estar atribuído a essa função. Isso significa que seu nome é usado como uma variável de saída.

Uma declaração de função inicia com a palavra-chave FUNCTION. Consultar: Recomendações sobre nomeação.

Em IL uma chamada de função só pode ser posicionada dentro de ações de um passo ou transição.

Exemplo de uma função Fct em IL, na qual três variáveis de entrada são declaradas. As duas primeiras são multiplicadas e depois divididas pela terceira. A função retorna o resultado dessa operação:

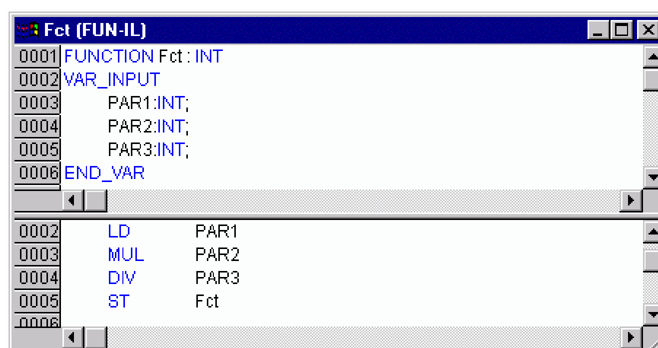


Figura 2-1. Exemplo de uma função em IL

Chamando uma função:

Em ST uma chamada de função pode ser usada como um operando em uma expressão.

Em SFC uma chamada de função só pode acontecer dentro de ações, de um passo ou transição.

Nota: o MasterTool IEC permite usar variáveis globais dentro de uma função. Isto intencionalmente desvia da norma IEC61131-3, a qual prescreve que o valor de retorno de uma função somente será modificado pelos parâmetros de entrada. Assim, a diferença entre funções e programas é justamente que a função retorna somente um valor de retorno e que seus parâmetros e valores de retorno são manipulados sobre a pilha.

Exemplos para chamar a função Fct descrita acima:

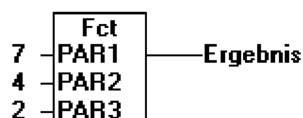
Em IL:

```
LD 7
FCT 2,4
ST ERGEBNIS
```

Em ST:

```
ERGEBNIS:= FCT (7, 2, 4);
```

Em FBD:



Atenção: se uma variável local é declarada como RETAIN, isto não terá nenhum efeito. A variável não será escrita na área retentiva!

Notas:

Se for definida uma função no projeto com o nome CheckBounds, ela pode ser usada para verificar overflows de intervalos. O nome da função é definido e pode ter somente este identificador.

Se forem definidas funções no projeto com os nomes CheckDivByte, CheckDivWord, CheckDivDWord e CheckDivReal, elas podem ser usadas para verificar o valor do divisor caso tenha sido usado o operador DIV, por exemplo, para evitar uma divisão por 0.

Se forem definidas funções com o nome CheckRangeSigned e CheckRangeUnsigned, então excessos nos intervalos das variáveis declaradas com tipos de subintervalos podem ser interceptados.

Todos estes nomes de função de verificação são reservados para o uso descrito.

Bloco Funcional

Um bloco funcional é uma POU que fornece um ou mais valores durante o seu uso.

Ao contrário da função, um bloco funcional não fornece nenhum valor de retorno.

Uma declaração de bloco funcional começa com a palavra-chave FUNCTION_BLOCK.

Reproduções ou instâncias (cópias) de um bloco funcional podem ser criadas.

Consultar: Recomendações sobre nomeação.

Exemplo em IL de um bloco funcional com duas variáveis de entrada e duas variáveis de saída. Uma saída é o produto das duas entradas; a outra é a comparação para a igualdade:

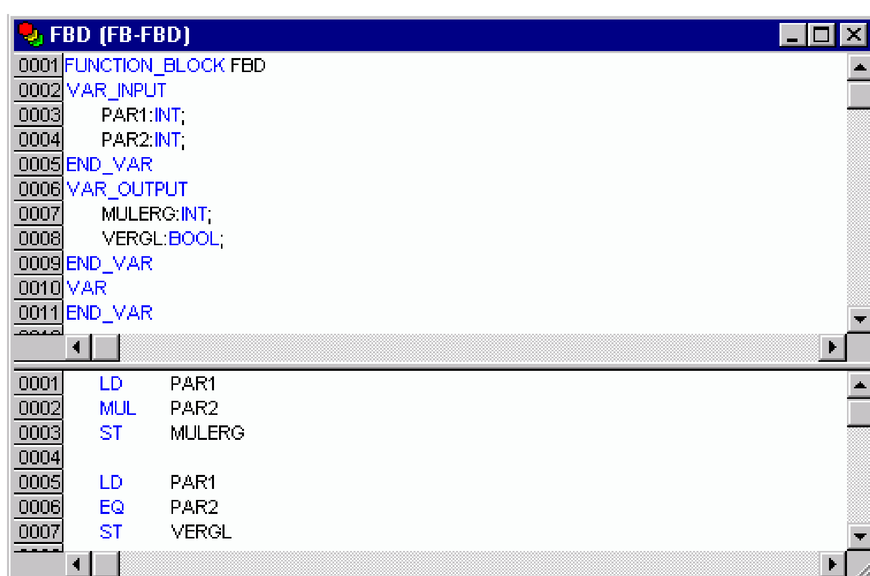


Figura 2-2. Exemplo em IL de um bloco funcional

Instâncias de blocos funcionais

Reproduções ou instâncias (cópias) de um bloco funcional podem ser criadas.

Cada instância possui seu próprio identificador (o nome da instância) e uma estrutura de dados que contém suas entradas, saídas e variáveis internas. As instâncias são declaradas localmente ou globalmente como variáveis, enquanto o nome do bloco funcional é indicado como o tipo de um identificador.

Consultar: Recomendações sobre nomeação.

Exemplo de uma instância com o nome da FUBINSTANCE do bloco funcional FUB:

```
FUBINSTANCE: FUB;
```

Blocos funcionais sempre são chamados por meio de instâncias descritas acima.

Somente os parâmetros de entrada e saída podem ser acessados de fora de uma instância de bloco funcional, já suas variáveis internas não podem.

Exemplo para acessar uma variável de entrada:

O bloco funcional FB tem uma variável de entrada in1 do tipo INT.

```
PROGRAM PROG

VAR
FBINST1:FB;
END_VAR

LD 17
ST FBINST1.IIN1
CAL FBINST1

END_PROGRAM
```

As partes de declaração dos blocos funcionais e dos programas podem conter declarações de instâncias. Declarações de instâncias não são permitidas em funções.

O acesso a uma instância de um bloco funcional é limitado à POU na qual ele foi declarado, a menos que ela tenha sido declarada globalmente.

O nome da instância de um bloco funcional pode ser usado como entrada para uma função ou um bloco funcional.

Nota: após o processamento de um bloco funcional, todos os valores são mantidos até o próximo bloco ser processado. Portanto, chamadas de bloco funcional com os mesmos argumentos nem sempre retornam os mesmos valores de saída!

Nota: se pelo menos uma das variáveis do bloco funcional for uma variável retentiva, o total da instância é armazenado na área retentiva.

Chamando um bloco funcional

As variáveis de entrada e saída de um bloco funcional podem ser acessadas a partir de outra POU configurando uma instância do bloco funcional, especificando a variável desejada e usando a seguinte sintaxe: <Nome da instância>. <Nome da variável>.

Atribuindo parâmetros na chamada

Caso deseja-se definir parâmetros de entrada e/ou saída quando chamar o bloco funcional, pode-se fazer isso nas linguagens de texto IL e ST, atribuindo valores aos parâmetros após o nome da instância do bloco funcional entre parênteses (para parâmetros de entrada essa atribuição acontece usando":=" assim como com a inicialização de variáveis na posição de declaração. Para os parâmetros de saída, deve ser usado "=>").

Se a instância é inserida por meio do Assistente de Entrada (<F2>) via opção **'Com argumentos'** na janela de implementação de uma POU ST ou IL, ela será exibida automaticamente conforme essa sintaxe com todos os seus parâmetros. Mas esses parâmetros não necessariamente precisam ser atribuídos.

Exemplo:

FBINST é uma variável local do tipo bloco funcional, que contém a variável entrada xx e a variável de saída yy. Quando FBINST é inserida em um programa ST por meio do Assistente de Entrada, a chamada será exibida como segue: FBINST1(XX:= , YY=>).

VAR_IN_OUT na chamada

Considerar que as VAR_IN_OUT de um bloco funcional são entregues como ponteiros. Por este motivo, em uma chamada de um bloco funcional nenhuma constante pode ser atribuída a elas e não há nenhuma leitura ou acesso de escrita (a partir de fora) nelas.

Exemplo:

```
VAR
FUBOINST: FUBO;
IVAR1: INT;
END_VAR
IVAR1:=2;
FUBOINST (IINOUT1:=IVAR1);
```

Não permitido neste caso: "fuboinst (iInOut1:=2);" ou "fuboinst.iInOut1:=2;"

Exemplos para chamar o bloco funcional FUB:

Bloco funcional FUB, consultar o capítulo 'Bloco funcional'.

O resultado da multiplicação é salvo na variável ERG e o resultado da comparação é salvo em QUAD. Uma instância de FUB com o nome INSTANCE é declarada.

A seguir está mostrado como a instância de um bloco funcional é chamada em IL:

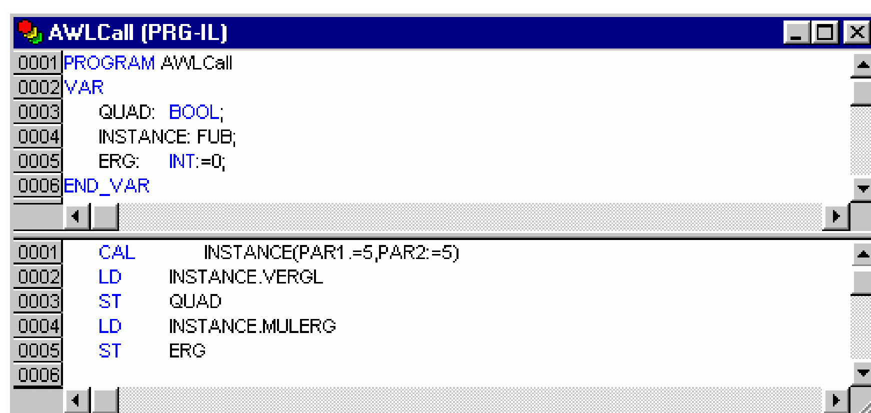


Figura 2-3. Como a instância de um bloco funcional é chamada em IL

A seguir está mostrado como a instância de um bloco funcional é chamada em ST (a parte de declaração é a mesma que em IL):

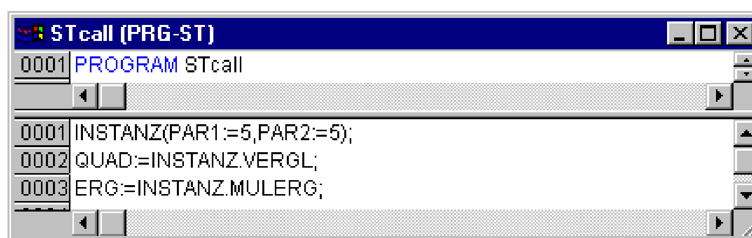


Figura 2-4. Como a instância de um bloco funcional é chamada em ST

A seguir está mostrado como a instância de um bloco funcional é chamada em FDB (a parte de declaração é a mesma que em IL):

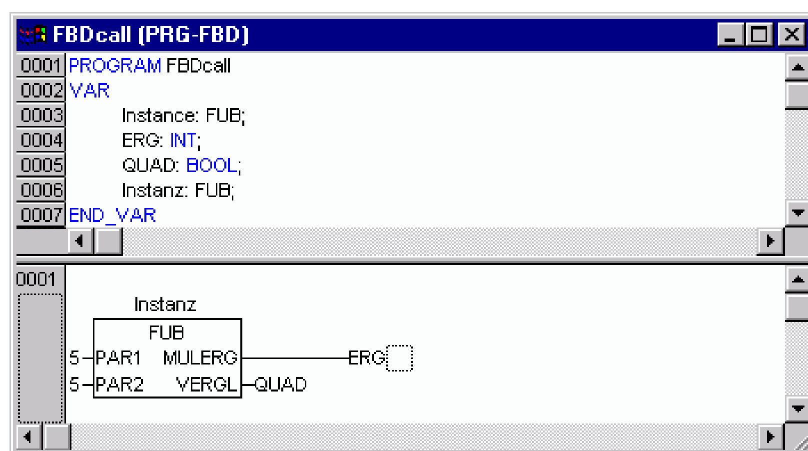


Figura 2-5. Como a instância de um bloco funcional é chamada em FDB

Nos blocos funcionais SFC, as chamadas só acontecem em passos.

Programa

Um programa é uma POU que retorna vários valores durante a operação. Programas são reconhecidos globalmente durante todo o projeto. Todos os valores são mantidos desde a última vez que o programa foi executado até a próxima execução.

Programas podem ser chamados. Uma chamada de programa em uma função não é permitida. Também não existem instâncias de programas.

Se uma POU chama um programa e, se assim, os valores dele são alterados, então, essas alterações serão mantidas na próxima vez que ele é chamado, mesmo se tiver sido chamado de dentro de outra POU.

Isso é diferente da chamada de um bloco funcional. Lá somente os valores na instância determinada de um bloco funcional são alterados.

Essas alterações, portanto, desempenham um papel somente quando a mesma instância é chamada.

Exemplo de um programa:

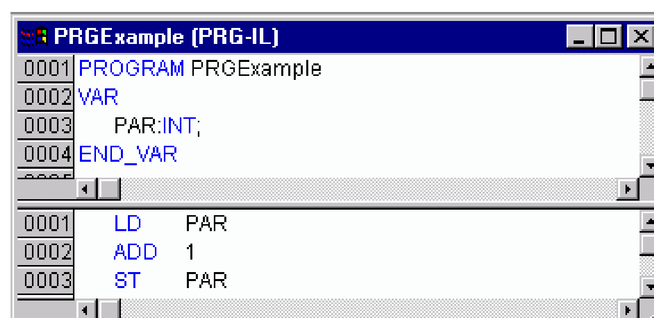


Figura 2-6. Exemplo de um programa

Uma declaração de programa começa com a palavra-chave PROGRAM e termina com END_PROGRAM. Considere as Recomendações sobre nomeação.

Para definir parâmetros de entrada e/ou saída quando da chamada do programa, pode-se fazer isso em linguagens de texto IL e ST, atribuindo valores aos parâmetros após o nome do programa entre parênteses (para parâmetros de entrada essa atribuição acontece usando":= " assim como na inicialização de variáveis na posição de declaração; para os parâmetros de saída deve ser usado"=>") .

Se o programa for inserido por meio do Assistente de Entrada (<F2>) via opção '**Com argumentos'** na janela de implementação de uma POU ST ou IL, ele será exibido automaticamente conforme essa

sintaxe com todos os seus parâmetros. Mas esses parâmetros não necessariamente precisam ser atribuídos.

Exemplos de chamadas de programa:

Em IL:

```
CAL PRGEXAMPLE2
LD PRGEXAMPLE2.OUT_VAR
ST ERG
```

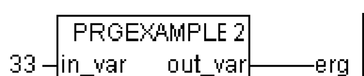
Ou atribuindo os parâmetros (assistente de entrada "Com argumentos", consultar acima):

```
CAL PRGEXAMPLE2 (IN_VAR:=33, OUT_VAR=>ERG )
```

EM ST:

```
PRGEXAMPLE2;
ERG:= PRGEXAMPLE2.OUT_VAR;
PRGEXAMPLE2 (IN_VAR:=33, OUT_VAR=>ERG );
```

Em FBD:



Exemplo de uma possível seqüência de chamada para PLC_PRG:

Consultar o PRG exemplo mostrado:

```
LD 0
ST PRG EXEMPLO.PAR (*CONFIGURAÇÃO PADRÃO PARA PAR É 0*)
CAL IL CALL (*ERG IN IL CHAMADA RESULTA EM 1*)
CAL ST CALL (*ERG IN ST CHAMADA RESULTA EM 2*)
CAL FBD CALL (*ERG IN FBD CHAMADA RESULTA EM 3*)
```

Se a variável PAR do programa PRG exemplo é inicializada pelo programa principal com (0) zero então os programas são chamados consecutivamente, o resultado ERG nos programas terá os valores 1, 2 e 3. Se um deles muda a seqüência de chamadas, então os valores dos parâmetros resultantes também se alteram de maneira correspondente.

PLC_PRG

O PLC_PRG é uma POU especial pré-definida. Cada projeto deve conter esse programa especial. Esta POU é chamada exatamente uma vez por ciclo de controle.

Na primeira vez que o comando 'Projeto' 'Acrescentar objetos' é usado, depois que um novo projeto é criado, a entrada padrão na caixa de diálogo POU será uma POU chamada PLC_PRG do tipo programa (PROGRAM). Não se deve alterar essa configuração padrão!

Se tarefas foram definidas, então talvez o projeto não contenha qualquer PLC_PRG, uma vez que, neste caso, a seqüência de procedimento depende da atribuição de tarefas.

> ATENÇÃO: **não exclua ou renomeie a POU PLC_PRG** (supondo que não se esteja usando uma configuração de tarefa)! O PLC_PRG geralmente é o programa principal em um programa de tarefa única.

Ação

Ações podem ser definidas e atribuídas a blocos funcionais e programas ('Projeto' 'Acrescentar ação'). A ação representa uma implementação adicional que pode ser totalmente criada em outra linguagem. Cada ação recebe um nome.

Uma ação opera com os dados a partir do bloco funcional ou a partir do programa ao qual ela pertence. A ação usa as mesmas variáveis de entrada/saída e locais que a implementação "normal".

Exemplo de uma ação em um bloco funcional:

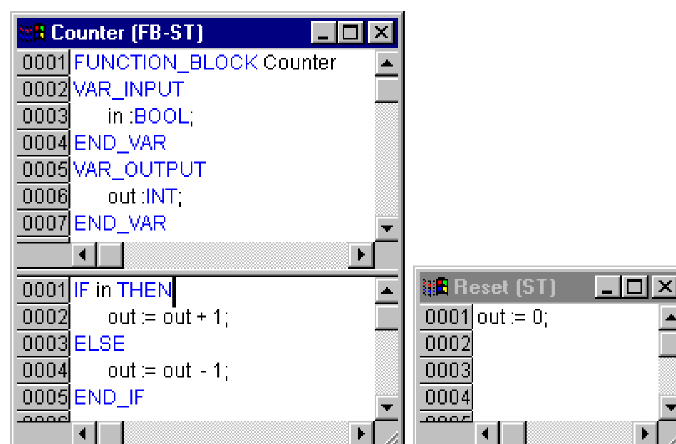


Figura 2-7. Exemplo de uma ação em um bloco funcional

No exemplo fornecido, a chamada do bloco funcional Counter aumenta ou diminui a variável de saída "out", dependendo do valor da variável de entrada "in". Chamando a ação Reset do bloco funcional, a variável de saída é levada para zero. A mesma variável "out" é escrita em ambos os casos.

Chamando uma ação

Uma ação é chamada com <Nome do programa>.<Nome da ação> ou <Nome da instância>. <Nome da ação>. Considere a notação no FBD (veja exemplo abaixo)! Se for necessário chamar a ação dentro de seu próprio bloco, simplesmente use o nome da ação em editores de texto e na forma gráfica. O bloco funcional chamará sem informações de instância.

Exemplos de chamadas das ações acima descritas a partir de outra POU:

Declaração para todos os exemplos:

```
PROGRAM PLC_PRG
VAR
INST: COUNTER;
END_VAR
```

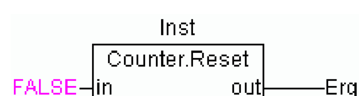
Chamada da ação 'Reset' em outra POU (programado em IL):

```
CAL INST.RESET (IN := FALSE)
LD INST. OUT
ST ERG
```

Chamada da ação 'Reset' em outra POU (programado em ST):

```
INST.RESET (IN := FALSE);
ERG: = INST.OUT;
```

Chamada da ação 'Reset' em outra POU (programado em FBD):



Nota: ações desempenham um papel importante em blocos no SFC. Consultar: Seqüenciamento Gráfico de Funções

A norma IEC não reconhece outras ações diferentes das do SFC.

Recursos

Os recursos são necessários para configurar e organizar o projeto e para rastreamento dos valores de variáveis:

- Variáveis globais que podem ser usadas em todo o projeto ou rede
- Gerenciador de Bibliotecas para adicionar as bibliotecas ao projeto
- LOG para registrar as ações durante uma sessão online
- Configuração de alarmes para a configuração do tratamento de alarmes no projeto
- Configuração do CP para configurar o hardware
- Configuração de tarefas para orientar seu programa através de tarefas
- Gerenciador de Monitoração e Receitas para exibir valores de variáveis e a configuração padrão dos valores das variáveis
- Configurações do dispositivo para a seleção e, se necessário, configuração final do sistema dispositivo
- Área de trabalho como uma imagem das opções de projeto

Dependendo do sistema do dispositivo e das configurações do dispositivo feitas no MasterTool IEC os recursos a seguir também podem estar disponíveis no seu projeto:

- Rastreamento de amostragem para exibição gráfica de valores de variáveis
- Gerenciador de parâmetros para troca de dados com outros controladores em uma rede
- Navegador - CP como monitor de controle
- Ferramentas – disponibilidade dependendo do dispositivo – para chamar programas de ferramentas externas a partir do MasterTool IEC.

Bibliotecas

Pode-se incluir no projeto uma série de bibliotecas cujas POUs, tipos de dados e variáveis globais podem ser usados como variáveis definidas pelo usuário. As bibliotecas Standard.lib e Util.lib são partes padrão do programa e estão sempre à disposição.

Consultar 'Gerenciador de Bibliotecas'.

Tipos de dados

Juntamente com os tipos de dados padrão, o usuário pode definir seus próprios tipos de dados. Estruturas, tipos de enumeração e referências podem ser criadas.

Consultar 'Tipos de dados'.

Visualização

O MasterTool IEC fornece telas de modo que se pode exibir as variáveis do projeto. Pode-se ainda plotar elementos geométricos offline com a ajuda da visualização. No modo online, elas podem alterar sua forma/cor/saída de texto em resposta a valores especificados das variáveis.

Uma tela pode ser usada como uma interface de operação pura para um CP com MasterTool IEC HMI ou com a Visualização do Dispositivo rodando diretamente no CP.

Para obter mais detalhes, consultar: 'Visualização'.

Linguagens

O MasterTool IEC oferece suporte a todas as linguagens descritas na norma IEC-61131.

Linguagens textuais:

- Lista de Instruções (IL)
- Texto Estruturado (ST)

Linguagens gráficas:

- Seqüenciamento Gráfico de Funções (SFC)
- Diagrama de Blocos Funcionais (FBD)
- Diagrama Ladder (LD)

Além disso, também está disponível - baseado no FBD – o Gráfico Contínuo de Funções -(CFC).

Lista de Instruções (IL)

Uma Lista de Instruções (IL) consiste em uma série de instruções. Cada instrução começa em uma nova linha e contém um operador e, dependendo do tipo de operação, um ou mais operandos separados por vírgulas.

Na frente de uma instrução pode haver uma marca de identificação (rótulo) seguida de dois pontos (:).

Um comentário deve ser o último elemento em uma linha. Linhas em branco podem ser inseridas entre instruções.

Exemplo:

```
LD 17
ST LINT (* COMENTÁRIO *)
GE 5
JMPC NEXT
LD IDWORD
EQ ISTRUCT.SDWORD
STN TEST
NEXT:
```

Modificadores e operadores em IL

Na linguagem IL os seguintes operadores e modificadores podem ser usados. Modificadores:

C	Com JMP, CAL, RET:	A instrução é executada somente se o resultado da expressão anterior for TRUE.
N	Com JMPC, CALC, RETC:	A instrução é executada somente se o resultado da expressão anterior for FALSE.
N	Caso contrário	Negação do operando (não do acumulador).

Tabela 2-1. Modificadores

Segue uma tabela com todos os operadores em IL, seus possíveis modificadores e significados relevantes:

Operador	Modificador	Significado
LD	N	Tornar o resultado atual igual ao do operando
ST	N	Salvar o resultado atual na posição do operando
S		Colocar o operando Booleano exatamente em TRUE se o resultado atual for TRUE
R		Colocar o operando Booleano exatamente em FALSE se o resultado atual for TRUE
AND	N,(	E lógico (bit a bit)
OR	N,(	OU lógico (bit a bit)
XOR	N,(	OU exclusivo lógico (bit a bit)
ADD	(	Adição
SUB	(	Subtração

MUL	(	Multiplicação
DIV	(	Divisão
GT	(	>
GE	(	>=
EQ	(	=
NE	(	<>
LE	(	<=
LT	(	<
JMP	CN	Saltar para o rótulo
CAL	CN	Chamar programa ou bloco funcional
RET	CN	Retornar a POU que chamou
)		Avaliar a operação adiada

Tabela 2-2. Operadores em IL

Exemplo de um programa em IL usando alguns modificadores:

```
LD      TRUE     (*Carregar TRUE no acumulador *)
ANDN    BOOL1    (* Executar E com o valor negado da variável BOOL1 *)
JMPC    mark     (*Se o resultado foi TRUE, então, saltar para o rótulo "mark" *)
LDN     BOOL2    (*salvar o valor negado de*)
ST      ERG      (*BOOL2 em ERG*)
mark:
LD      BOOL2    (*salvar o valor de*)
ST      ERG      (*BOOL2 em ERG*)
```

Também é possível em IL colocar parênteses após uma operação. O valor do parêntese é, então, considerado como um operando.

Exemplo:

```
LD 2
MUL 2
ADD 3
ERG
```

O valor de ERG é 7.

No entanto, se utilizarmos parênteses:

```
LD 2
MUL (2
ADD 3
)
ST ERG
```

O valor resultante para Erg será 10, a operação MUL será avaliada somente no")"; o operando para MUL 5, será então calculado.

Texto Estruturado (ST)

O Texto Estruturado consiste em uma série de instruções que, como determinado em linguagens de alto nível, ("IF..THEN..ELSE") ou em laços (WHILE..DO) podem ser executadas.

Exemplo:

```
IF VALUE < 7 THEN
WHILE VALUE < 8 DO
VALUE:=VALUE+1;
END_WHILE;
```

END_IF;

Expressões

Uma expressão é uma construção que retorna um valor após a sua avaliação.

Expressões são compostas de operadores e operandos. Um operando pode ser uma constante, uma variável, uma chamada de função ou outra expressão.

Avaliação de expressões

A avaliação de uma expressão ocorre por meio do processamento dos operadores de acordo com certas regras de precedência. O operador com a maior precedência é processado primeiro, em seguida, o operador com a próxima precedência maior, e assim por diante, até que todos os operadores sejam processados.

Operadores com mesma ordem de precedência são processados da esquerda para a direita.

Segue uma tabela de operadores ST na ordem de precedência:

Operação	Símbolo	Característica
Entre parênteses	(expressão)	Maior precedência
Chamada de função	Nome da função (lista de parâmetros)	
Exponencial	EXPT	
Negação (complemento)	- (NOT)	
Multiplicação **Divisão** **Módulo**	* / MOD	
Adição **Subtração**	+ -	
Comparação	<,>,<=,>=	
Igual a **Diferente de**	= <>	
E Booleano	AND	
OU exclusivo Booleano	XOR	
OU Booleano	OR	Menor precedência

Tabela 2-3. Operadores ST

A seguir, exemplos de algumas instruções em ST:

Atribuição:
```
A:=B; CV:= CV + 1; C:=SIN(X);
```

Chamando um bloco funcional e usando uma saída FB:
```
CMD_TMR (IN := %IX5, PT := 300);
A: =CMD_TMR.Q;
```

RETURN:
```
RETURN;
```

IF:
```
D:=B*B;
IF D<0.0 THEN
C: =A;
ELSIF D=0.0 THEN
C: =B;
ELSE
C: =D;
END_IF;
```

CASE:
```
CASE INT1 OF
1: BOOL1:= TRUE;
2: BOOL2:= TRUE;
ELSE
 BOOL1:= FALSE;
 BOOL2:= FALSE;
END_CASE;
```

FOR:
```
J: =101;
FOR I: =1 TO 100 BY 2 DO
IF ARR [I] = 70 THEN
```

	J: =I; EXIT; END_IF; END_FOR;
WHILE:	J: =1; WHILE J<= 100 AND ARR [J] <> 70 DO J: =J+2; END_WHILE;
REPEAT:	J: =-1; REPEAT J: =J+2; UNTIL J= 101 OR ARR[J] = 70 END_REPEAT;
EXIT:	EXIT;
Instrução vazia:	;

Operador de atribuição

No lado esquerdo de uma atribuição, há um operando (variável, endereço) ao qual está atribuído o valor da expressão, no lado direito, com o operador de atribuição:=.

Exemplo:

```
VAR1:= VAR2 * 10;
```

Após a conclusão desta linha Var1 tem o valor de Var2 multiplicado por 10.

Chamando blocos funcionais em ST

Um bloco funcional é chamado em ST escrevendo o nome da instância do bloco funcional e, então, atribuindo os valores dos parâmetros entre parênteses. No exemplo a seguir, um temporizador é chamado com atribuições para os parâmetros IN e PT. Em seguida, a variável de resultado Q é atribuída à variável A.

A variável resultante, como em IL, é endereçada com o nome do bloco funcional seguido de um ponto e o nome da variável:

```
CMD_TMR (IN: = %IX5, PT: = 300);
A:= CMD_TMR. Q
```

Instrução RETURN

A instrução RETURN pode ser usada para deixar uma POU, por exemplo, dependendo da condição.

Instrução IF

Com a instrução IF pode-se verificar uma condição e, dependendo desta condição, executar instruções.

Sintaxe:

```
IF <EXPRESSÃO BOOLEANA1> THEN
<IF_INSTRUÇÕES>
{ELSIF <EXPRESSÃO BOOLEANA2> THEN
<ELSIF_INSTRUÇÕES1>
.
.
ELSIF <EXPRESSÃO BOOLEANA N> THEN
<ELSIF_INSTRUÇÕES N-1>
ELSE
<ELSE_INSTRUÇÕES>}
END_IF;
```

A parte entre chaves {} é opcional.

Se a <Expressão booleana1> retorna TRUE, então apenas as < IF_Instruções > são executadas.

Caso contrário, as expressões booleanas começando com <Expressão booleana2>, são avaliadas uma após a outra até que uma das expressões retorne TRUE. Então, somente essas instruções, após essa expressão booleana e antes do próximo ELSE ou ELSIF são avaliadas.

Se nenhuma das expressões booleanas produz TRUE, somente as < ELSE_Instruções > são avaliadas.

Exemplo:

```
IF TEMP<17
THEN HEATING_ON:= TRUE;
ELSE HEATING_ON:= FALSE;
END_IF;
```

Aqui o aquecimento é ativado quando a temperatura dos coletores for abaixo de 17 graus. Caso contrário, ele permanece desativado.

Instrução CASE

Com as instruções CASE é possível combinar várias instruções condicionadas com a mesma variável de condição em uma construção.

Sintaxe:

```
CASE <VAR1> OF
<VALUE1>: <INSTRUÇÃO 1>
<VALUE2>: <INSTRUÇÃO 2>
<VALUE3, VALUE4, VALUE5>: <INSTRUÇÃO 3>
<VALUE6 . VALUE10>: <INSTRUÇÃO 4>
...
<VALUE N>: <INSTRUÇÃO N>
ELSE <ELSE INSTRUÇÃO>
END_CASE;
```

Uma instrução CASE é processada de acordo com o seguinte modelo:

Se a variável em <Var1> tiver o valor <Valor i>, então a instrução <Instrução i> será executada.

Se <Var1> não tem nenhum dos valores indicados, então <ELSE Instrução> será executada.

Se a mesma instrução está para ser executada em diversos valores das variáveis, então, esses valores são escritos um após o outro, separados por vírgulas e, em seguida, a condição da execução comum.

Se a mesma instrução está para ser executada em um intervalo de valor de uma variável, escreve-se o valor inicial e o final, separados por dois pontos, um após o outro. Na seqüência vem a condição comum.

Exemplo:

```
CASE INT1 OF
1, 5: BOOL1:= TRUE;
BOOL3:= FALSE;
2: BOOL2:= FALSE;
BOOL3:= TRUE;
10.20: BOOL1:= TRUE;
BOOL3:= TRUE;
```

```
ELSE
BOOL1:= NOT BOOL1;
BOOL2:= BOOL1 OR BOOL2;
END_CASE;
```

Laço FOR

Com o laço FOR é possível programar processos repetidos.

Sintaxe:

```
INT_VAR :INT;
FOR <INT_VAR> := <INIT_VALUE> TO <END_VALUE> {BY <STEP SIZE>} DO
<INSTRUÇÕES>
END_FOR;
```

A parte entre chaves {} é opcional.

As <Instruções> são executadas desde que o contador <INT_Var> não seja maior do que o <END_VALUE>. Isso é verificado antes de executar as <Instruções> para que as <Instruções> nunca sejam executadas se <INIT_VALUE> for maior do que <END_VALUE>.

Quando <Instruções> são executadas, <INT_Var> sempre é incrementado em <Step size>. O tamanho da etapa pode ter qualquer valor inteiro. Se estiver faltando, então ele é definido para 1. O laço também deve terminar se <INT_Var> ficar maior.

Exemplo:

```
FOR COUNTER:=1 TO 5 BY 1 DO
VAR1:=VAR1*2;
END_FOR;
ERG:=VAR1;
```

Vamos presumir que a configuração padrão para Var1 é o valor 1. Então, ele terá o valor 32 após o laço FOR.

Nota: <END_VALUE> não deve ser igual ao valor limite do contador <INT_VAR>. Por exemplo: se a variável Counter é do tipo SINT e se <END_VALUE> é 127, ocorre um laço infinito.

Laço WHILE

O laço WHILE pode ser usado como o laço FOR, com a diferença de que a condição de interrupção pode ser qualquer expressão booleana. Isso significa que é indicada uma condição que, quando atendida, executará o laço.

Sintaxe:

```
WHILE <EXPRESSÃO_BOOLEANA>
<INSTRUÇÕES>
END_WHILE;
```

As <Instruções> são executadas repetidamente, desde que a <Expressão_booleana> retorne TRUE. Se a <Expressão_booleana> já for FALSE na primeira avaliação, então as <Instruções> nunca são executadas. Se <Expressão_booleana> nunca assumir o valor FALSE, então as <Instruções> serão repetidas indefinidamente, o que causa um retardo relativo.

Nota: o programador deve certificar-se de que nenhum laço sem fim é gerado. Ele faz isso, alterando a condição na parte do laço, por exemplo, via contador bidirecional.

Exemplo:

```
WHILE COUNTER<>0 DO
VAR1:= VAR1*2;
COUNTER:= COUNTER-1;
END_WHILE
```

Os laços WHILE e REPEAT são, em um determinado sentido, mais eficientes do que o laço FOR, uma vez que não precisam saber o número de ciclos antes de executar o laço. Em certos casos é possível trabalhar com esses dois tipos de laço. Se, no entanto, o número de ciclos de laço é claro, então um laço FOR é preferível, pois ele não permite laço sem fim.

Laço REPEAT

O laço REPEAT é diferente do laço WHILE, porque a condição de interrupção é verificada somente após ele ter sido executado. Isso significa que será executado pelo menos uma vez, independentemente do texto da condição de interrupção.

Sintaxe:

```
REPEAT
<INSTRUÇÕES>
UNTIL < EXPRESSÃO_BOOLEANA >
END_REPEAT;
```

As <Instruções> são realizadas até que a <Expressão_booleana> retorne TRUE.

Se <Expressão booleana> produzir já na primeira avaliação TRUE, então as <Instruções> são executadas apenas uma vez. Se <Expressão_booleana> nunca assumir o valor TRUE, então, as <Instruções> serão repetidas indefinidamente, o que causa um retardo relativo.

Nota: o programador deve certificar-se de que nenhum laço sem fim é causado. Ele faz isso, alterando a condição na parte da instrução do laço, por exemplo, acionando um contador progressivo ou regressivo.

Exemplo:

```
REPEAT
VAR1:= VAR1*2;
COUNTER: = COUNTER-1;
UNTIL
COUNTER=0
END_REPEAT;
```

Instrução EXIT

Se a instrução EXIT aparece em um laço FOR, WHILE ou REPEAT, então, o laço mais interno é encerrado, independentemente da condição de interrupção.

Seqüenciamento Gráfico de Funções (SFC)

O Seqüenciamento Gráfico de Funções (SFC) é uma linguagem orientada graficamente que torna possível descrever a ordem cronológica de diferentes ações dentro de um programa. Para isso, as ações são atribuídas aos elementos do passo e a seqüência de processamento é controlada pelos elementos de transição.

Exemplo de uma rede em SFC:

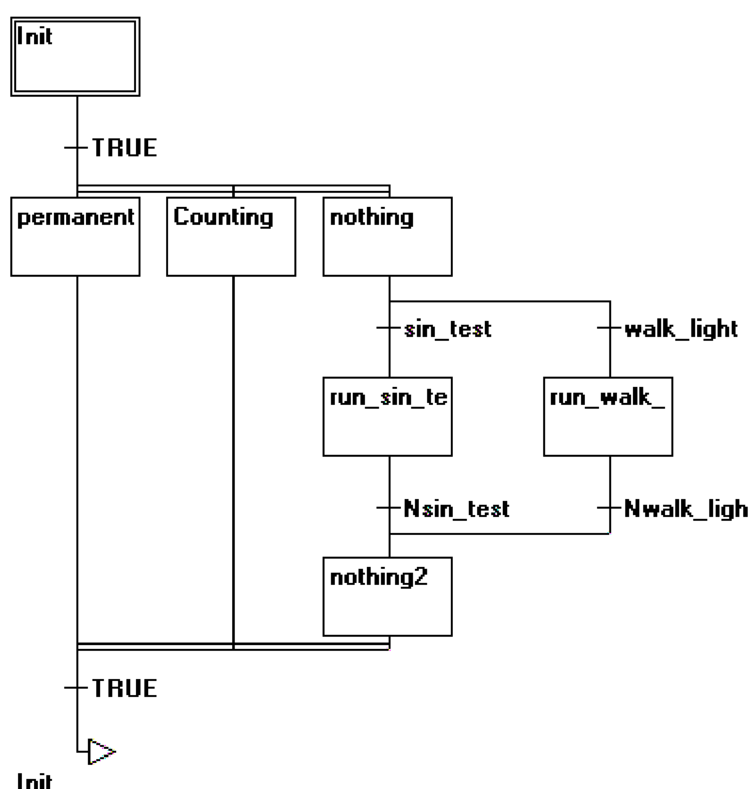


Figura 2-8. Exemplo de uma rede em SFC

Para mais informações sobre o Editor SFC consultar o capítulo 5.

Passo

Uma POU escrita em SFC consiste em uma série de passos que são conectados entre si através de conexões direcionadas (transições).

Há dois tipos de passos:

- O tipo simplificado consiste em uma ação e um sinalizador que mostra se o passo está ativo. Se a ação de um passo for implementada, então aparecerá um pequeno triângulo no canto superior direito do passo.
- Um passo IEC consiste de um sinalizador e uma ou mais ações atribuídas ou variáveis booleanas. As ações associadas são exibidas à direita do passo.

Ação

Uma ação pode conter uma série de instruções em IL ou em ST, muitas redes em FBD ou LD ou mesmo gráficos SFC

Com os passos simplificados uma ação sempre está conectada a um passo. Para editar uma ação, clique duas vezes com o mouse sobre o passo ao qual a ação pertence. Ou selecione o passo e selecione o comando do menu 'Extras' 'Zoom Ação/Transição'. Adicionalmente, uma ação de entrada ou saída por passo é possível.

Ações de passos IEC aparecem diretamente no Organizador de objetos sob a sua POU SFC e são carregadas com um clique duplo ou pressionando <Enter> no editor. Novas ações podem ser criadas com o 'Projeto' 'Acrescentar ação'. Pode-se atribuir, no máximo, nove ações para um passo IEC.

Ação de Entrada ou Saída

Além da ação de passo, pode-se adicionar uma ação de entrada e uma ação de saída a um passo. Uma ação de entrada é executada apenas uma vez, logo após o passo ser ativado. Uma ação de saída é executada apenas uma vez antes que o passo seja desativado.

Um passo com ação de entrada é indicado por um "E" no canto inferior esquerdo e a ação de saída por um "X" no canto inferior direito.

A ação de entrada e saída pode ser implementada em qualquer linguagem. Para editar uma ação de entrada ou saída, dê um clique duplo com o mouse no canto correspondente do passo.

Exemplo de um passo com ação de entrada e saída:

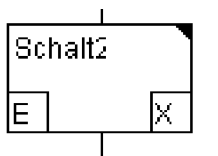


Figura 2-9. Exemplo de um passo com ação de entrada e saída

Transição/Condição de transição

Entre os passos existem transições.

Uma condição de transição deve ter o valor TRUE ou FALSE. Portanto, ela pode consistir tanto de uma variável booleana, quanto de um endereço booleano ou de uma constante booleana. Ela também pode conter uma série de instruções com um resultado booleano, tanto na sintaxe ST (ex.: (i<= 100) AND b) ou em qualquer linguagem desejada (consultar 'Extras' 'Zoom Ação/Transição'). Entretanto, uma transição não pode conter programas, blocos funcionais ou atribuições!

No Editor SFC, uma condição de transição pode ser escrita diretamente no símbolo de transição ou uma janela do próprio editor pode ser aberta para inserir a condição extra ('Extras' 'Zoom Ação/Transição'). Considere que as instruções inseridas na janela do editor terão precedência!

Nota: além de transições, um modo específico também pode ser usado para pular para o próximo passo. Consultar o capítulo 5 para mais informações.

Passo ativo

Depois de chamar a POU SFC, a ação (circundada por uma borda dupla) pertencente ao passo inicial é executada primeiro. Um passo cuja ação está sendo executada é chamado ativo. No modo online os passos ativos são mostrados em azul.

Em um ciclo de controle, todas as ações que pertencem aos passos ativos são executadas. Daí em diante, os passos seguintes se tornam ativos se as condições de transição dos passos a seguir forem TRUE. Os passos ativos, no momento, serão executados no próximo ciclo.

Nota: se o passo ativo contém uma ação de saída, isso será executado apenas durante o próximo ciclo, desde que a transição seguinte seja TRUE.

Nota: considere a possibilidade de usar variáveis implícitas para armazenar o status e o tempo dos passos e ações.

Juntamente com os passos simplificados, também estão disponíveis no SFC os passos no padrão IEC.

Para ser capaz de usar passos IEC, deve-se vincular a biblioteca especial SFC iecsfc.lib no projeto.

Não mais do que nove ações podem ser atribuídas a um passo IEC. Ações IEC não são fixas como ações de entrada, passo ou ações de saída para um determinado passo, como nos passos simplificados, mas são armazenadas separadamente a partir dos passos e podem ser reutilizadas muitas vezes dentro de uma POU. Para isso devem ser associadas aos passos únicos com o comando 'Extras' 'Associar ação'.

Junto com ações, variáveis booleanas podem ser atribuídas aos passos.

A ativação e desativação de ações e variáveis booleanas podem ser controladas usando os chamados qualificadores. Atrasos de tempo são possíveis. Desde que uma ação ainda possa estar ativa, se o

próximo passo foi processado, por exemplo, por meio do qualificador S (set), uma ação pode alcançar processos concorrentes.

Uma variável booleana associada é definida ou ressetada a cada chamada do bloco SFC. Isso significa que, a cada chamada a variável fica com o valor TRUE ou FALSE atribuído.

As ações associadas a um passo IEC são mostradas à direita do passo em uma caixa de duas partes. O campo esquerdo contém o qualificador, possivelmente com constante de tempo, e o campo direito contém o nome da ação, respectivamente, nome da variável booleana.

Um exemplo de um passo IEC com duas ações:

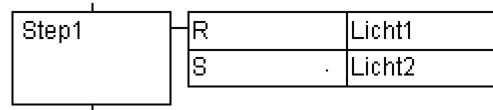


Figura 2-10. Exemplo de um passo IEC com duas ações

Para seguir os processos mais facilmente, todas as ações ativas no modo online são mostradas em azul como os passos ativos. Após cada ciclo é feita uma verificação para ver quais ações estão ativas.

Preste atenção também às restrições no uso dos qualificadores de tempo nas ações que são repetidamente reutilizadas dentro do mesmo ciclo (consultar 'Qualificador')!

Nota: se uma ação foi desativada, ela será executada mais uma vez. Isso significa que, cada ação é executada pelo menos duas vezes (também uma ação com o qualificador P).

No caso de uma chamada, primeiro são executadas as ações desativadas e, em seguida, as ações ativas, em ordem alfabética, a cada vez.

Se um passo recém-inserido é um passo IEC, dependerá se o comando do menu **'Extras' 'Usar passos IEC'** foi escolhido.

No Organizador de objetos, as ações aparecem diretamente sob suas respectivas POUs SFC. Novas ações podem ser criadas com 'Projeto' 'Acrescentar ação'.

Para usar passos IEC deve-se incluir no projeto a biblioteca especial SFC Iecsfc.lib.

POU SFC com ações no Organizador de objetos:

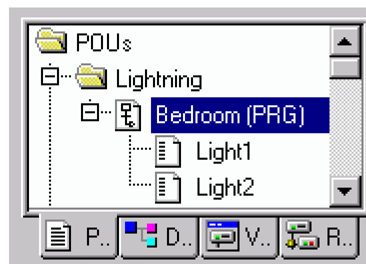


Figura 2-11. POU SFC com ações no Organizador de objetos

Qualificador

Os seguintes qualificadores estão disponíveis para associar as ações com os passos IEC:

- N (Não-armazenado) - A ação está ativa enquanto o passo estiver ativo;
- R (Reset) - A ação é desativada;
- S (Set) - A ação é ativada e permanecerá ativa até haver um reset;
- L (Tempo limitado) - A ação é ativada por um determinado tempo (no máximo enquanto o passo estiver ativo);
- D (Retardo de tempo) - A ação ficará ativa após um determinado período de tempo se a etapa ainda estiver ativa e, então, ela permanecerá ativa desde que o passo esteja ativo;
- P (Pulso) - A ação é executada apenas uma vez se o passo estiver ativo;

- SD (Armazenados e retardados no tempo) - A ação é ativada após um determinado período de tempo e permanecerá ativa até um reset;
- DS (Retardados e armazenados) - A ação é ativada após um determinado tempo, contanto que o passo ainda esteja ativo e permanecerá ativa até um reset;
- SL (Armazenados e limitados no tempo) - A ação é ativada por um determinado tempo.

Os qualificadores L, D, SD, DS e SL necessitam de um valor de tempo no formato de constante de TEMPO. Esse valor pode ser inserido diretamente (ex.: “T#5s”) ou por uma variável do tipo de dados TIME (ex.: “t_var”).

Nota: quando uma ação for desativada, ela será executada mais uma vez. Isso significa que cada ação é executada ao menos duas vezes (também uma ação com o qualificador P).

Variáveis implícitas no SFC

No SFC, variáveis declaradas implicitamente ("sinalizadores") podem ser usadas para verificar o status de passos e ações, bem como o tempo dos passos. Esses sinalizadores estão definidos no início de um ciclo. Para passos IEC e ações IEC, eles são fornecidos pela biblioteca iecsfc.lib (estruturas SFCStepType e SFCActionType), as quais estão automaticamente incluídas em um projeto.

Verificação do status do passo ou ação por meio de variáveis booleanas:

- Para passos IEC: <nomedopasso>.x e <nomedopasso>._x: <nomedopasso>.x mostra o status de ativação atual. <nomedopasso>._x mostra o status de ativação para o próximo ciclo. Se <nomedopasso>.x = TRUE o passo será executado no ciclo atual. Se <nomedopasso>._x = TRUE e <nomedopasso>.x = FALSE, o passo será executado no ciclo seguinte, isto é, <nomedopasso>._x é copiada para <nomedopasso>.x no início de um ciclo.
- Para passos simplificados: <nomedopasso> e _ <nomedopasso>: se <nomedopasso> = TRUE, o passo será executado no ciclo atual. Se _ <nomedopasso> = TRUE, o passo será executado no ciclo seguinte, isto é, <nomedopasso> é copiado para _ <nomedopasso> no início de um ciclo.
- Para ações IEC: <nomedaação>.x obtém TRUE assim que a ação é ativada (<nomedaação>._x é apenas para fins internos, não para uma verificação de status).

Tempo de um passo por meio de variáveis de TEMPO:

As seguintes variáveis implícitas fornecem o intervalo de tempo atual decorrido desde que o passo foi ativado; elas servem apenas para os passos que têm um tempo mínimo configurado nos atributos de passo.

- Para passos IEC: <nomedopasso>.t (<nomedopasso>._t não pode ser usado para fins externos)
- Para passos simplificados: _time <nomedopasso>. MAS: se essa variável implícita for usada para fins de verificação, ela também deve ser declarada explicitamente como uma variável de tempo; por exemplo, "_timeStep1: TIME;"
- Para ações IEC: as variáveis de tempo implícitas não são usadas.

Esses sinalizadores de status podem ser usados em cada ação e transição do módulo SFC. Mas eles também podem ser acessados de outros programas:

Exemplo: boolvar1:=sfc.step1.x;

Step1.x neste exemplo é uma variável booleana implícita mostrando o status do passo IEC "step1" na POU "sfc1".

Sinalizadores SFC

Para controlar a operação de POUs SFC, sinalizadores podem ser usados, os quais são criados implicitamente durante a execução do projeto. Para ler esses sinalizadores é necessário definir variáveis globais ou locais apropriadas como entradas ou saídas. Exemplo: se em uma POU SFC um passo está ativo por mais tempo do que definido em seus atributos, então um sinalizador será definido, o qual estará acessível usando uma variável "SFCError" (SFCError fica TRUE neste caso).

As seguintes variáveis de sinalizador podem ser definidas:

- **SFCEnableLimit**: essa variável é do tipo BOOL. Quando seu valor é TRUE, os timeouts dos passos serão registrados no SFCError. Outros timeouts serão ignorados.
- **SFCInit**: quando essa variável booleana tem o valor TRUE, o SFC retorna para o passo de inicialização. Os outros sinalizadores do SFC são zerados também (inicialização). O passo de inicialização permanece ativo, mas não é executado, enquanto a variável tiver o valor TRUE. Somente quando SFCInit é novamente definida como FALSE é que o bloco poderá ser processado normalmente.
- **SFCReset**: essa variável, do tipo BOOL, comporta-se de forma semelhante à SFCInit. Diferentemente desta, no entanto, processamento adicional ocorre após a inicialização do passo "Init". Portanto, por exemplo, o sinalizador SFCReset poderia ser redefinido para FALSE no passo de inicialização.
- O SFCReset também pode ser usado para redefinir ações booleanas associadas aos passos IEC - o que não era possível antes.
- **SFCQuitError**: neste caso a execução do SFC é interrompida desde que essa variável booleana tenha o valor TRUE . Todos os timeouts nos passos ativos são zerados quando a variável novamente assume o valor FALSE. É uma pré-condição que o sinalizador SFCError tenha sido definido também, o qual registra qualquer timeout no SFC.
- **SFCPause**: a execução do diagrama SFC é interrompida enquanto essa variável booleana tiver valor TRUE.
- **SFCError**: essa variável booleana é TRUE quando ocorrer um timeout em um diagrama SFC. Se outro ocorrer em um programa após o primeiro, ele não será registrado, a menos que a variável SFCError seja zerada primeiro. É um pré-requisito que SFCError seja definido, caso se deseje usar os outros sinalizadores de controle de tempo (SFCErrorStep SFCErrorPOU, SFCQuitError, SFCErrorAnalyzation).
- **SFCTrans**: essa variável booleana fica com valor TRUE quando uma transição é acionada.
- **SFCErrorStep**: essa variável é do tipo STRING. Se SFCError registrar um timeout, nesta variável é armazenado o nome do passo que provocou o timeout. É um pré-requisito que o sinalizador SFCError também tenha sido definido, o qual registra qualquer timeout no SFC.
- **SFCErrorPOU**: essa variável do tipo STRING contém o nome do bloco no qual ocorreu um timeout. É um pré-requisito que o sinalizador SFCError também tenha sido definido, o qual registra qualquer timeout no SFC.
- **SFCCurrentStep**: essa variável é do tipo STRING. O nome do passo que está ativo é armazenado nessa variável, independentemente do tempo de monitoração. No caso de seqüências simultâneas o passo armazenado é o da ramificação direita externa. Nenhum timeout a mais será registrado se ocorrer um timeout e a variável SFCError não for zerada novamente.
- **SFCErrorAnalyzationTable**: essa variável do tipo ARRAY [0..n] (resultante da expressão) fornece o resultado de uma análise de uma expressão de transição. Para cada componente da expressão, que está contribuindo para um FALSE da transição e, portanto, para um timeout do passo anterior, as seguintes informações são gravadas na estrutura ExpressionResult: nome, endereço, comentário, valor atual.
  Isso é possível para, no máximo, 16 componentes (variáveis), assim, o intervalo de matriz é no máximo 0...15.

  A estrutura, ExpressionResult, bem como os módulos de análise usados implicitamente, são fornecidos pela biblioteca AnalyzationNew.lib. Os módulos de análise também podem ser usados explicitamente em outras POUs, que não são programados em SFC.

  É um pré-requisito para a análise de uma expressão de transição, que um timeout seja registrado no passo anterior. Portanto, uma monitoração de tempo deve ser implementada e também a variável SFCError (veja acima) deve ser definida na janela de declaração.

- **SFCTip, SFCTipMode**: estas variáveis do tipo BOOL habilitam o modo "inching" do SFC. Quando SFCTipMode = TRUE só é possível pular para o próximo passo se SFCTip é definida como TRUE. Desde que SFCTipMode seja definida como FALSE, é possível pular até mesmo sobre as transições.

Nota: considere também as variáveis implícitas usadas para mapear o status e o tempo dos passos e ações.

Ramificação alternativa

Duas ou mais ramificações no SFC podem ser definidas como ramificações alternativas. Cada ramificação alternativa deve começar e terminar com uma transição. Ramificações alternativas podem conter ramificações paralelas e outras ramificações alternativas. Uma ramificação alternativa começa em uma linha horizontal (início alternativo) e termina em uma linha horizontal (final alternativo) ou com um salto.

Se um passo que precede a linha alternativa de início estiver ativo, então, a primeira transição de cada ramificação alternativa é avaliada da esquerda para a direita. A primeira transição a partir da esquerda, cuja condição de transição tem o valor TRUE, é avaliada e os passos a seguir são ativados (consultar passos ativos).

Ramificação paralela

Duas ou mais ramificações no SFC podem ser definidas como ramificações paralelas. Cada uma delas deve começar e terminar com um passo. Elas podem conter ramificações alternativas ou outras ramificações paralelas. Uma ramificação paralela começa com uma linha dupla (início paralela) e termina com uma linha dupla (final paralela) ou com um salto. Ela pode ser fornecida com um rótulo de salto.

Se a linha de início paralela do passo anterior está ativa e a condição de transição após esse passo tiver o valor TRUE, então os primeiros passos de todas as ramificações paralelas ficam ativos (consultar passo ativo). Essas ramificações agora são processadas paralelamente umas às outras. O passo posterior à linha paralela final se torna ativo quando todos os passos anteriores estiverem ativos e a condição de transição anterior a este passo produzir o valor TRUE.

Salto

Um salto é uma conexão para o passo cujo nome é indicado sob o símbolo do salto. Saltos são necessários porque não é permitido criar conexões que levem para cima ou que se cruzem entre si.

Diagrama de Blocos Funcionais (FBD)

O Diagrama de Blocos Funcionais é uma linguagem de programação orientada graficamente. Ele funciona com uma lista de redes na qual cada uma (rede) contém uma estrutura que representa tanto uma expressão lógica quanto aritmética, a chamada de um bloco funcional, um salto ou ainda uma instrução de retorno.

Exemplo de uma rede no FBD:

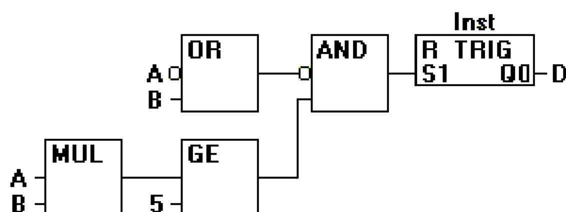


Figura 2-12. Exemplo de uma rede no FBD

Para mais informações sobre o editor FBD consultar o capítulo 5.

Gráfico Contínuo de Funções (CFC)

O Gráfico Contínuo de Funções (CFC) não funciona com redes, mas com elementos livremente posicionáveis. Isso permite, por exemplo, implementar realimentação.

Para mais informações sobre o editor CFC consultar o capítulo 5.

Exemplo de uma rede no editor CFC:

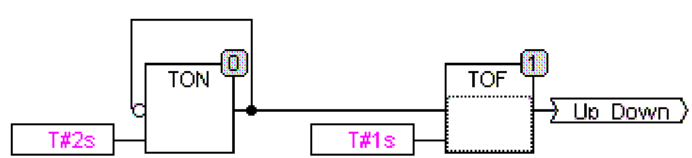


Figura 2-13. Exemplo de uma rede no editor CFC

Diagrama Ladder (LD)

O Diagrama Ladder (LD) também é uma linguagem de programação orientada a elementos gráficos que se aproxima da estrutura de um circuito elétrico.

Por um lado, o Diagrama Ladder (LD) é adequado para construir intertravamentos lógicos, por outro lado, também pode criar redes em FBD. Portanto o LD é muito útil para controlar a chamada de outras POUs.

O Diagrama Ladder (LD) consiste em uma série de redes. Uma rede é limitada nos lados direito e esquerdo por uma linha vertical. No meio está um diagrama de circuito composto de contatos, bobinas e linhas de conexão.

Cada rede consiste no lado esquerdo de uma série de contatos que passam a condição "On" ou "Off" da esquerda para a direita, os quais correspondem a valores booleanos TRUE e FALSE. A cada contato está associada uma variável booleana. Se essa variável é TRUE, a condição será passada da esquerda para a direita ao longo da linha de conexão. Caso contrário, a conexão à direita recebe o valor OFF.

Exemplo de uma rede em um Diagrama Ladder (LD) formado por contatos e bobinas:

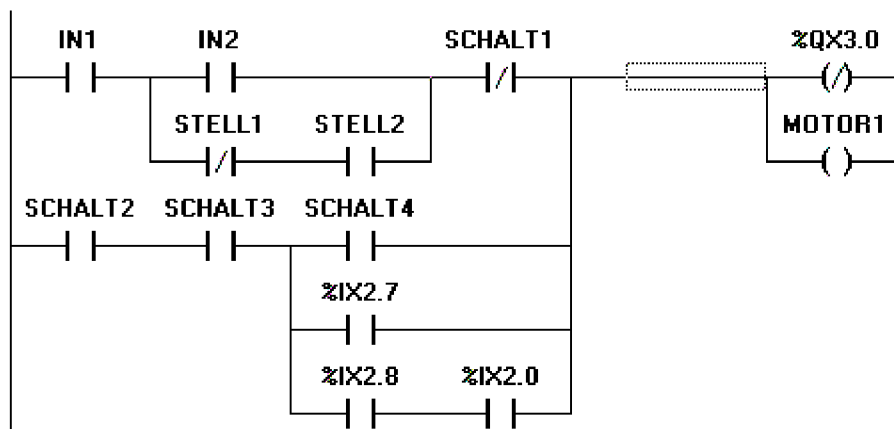


Figura 2-14. Exemplo de uma rede em um Diagrama Ladder

Para mais informações sobre o editor Ladder consultar o capítulo 5.

Contato

Cada rede no LD consiste, no lado esquerdo, de uma rede de contatos (contatos são representados por duas linhas paralelas: | |) que, da esquerda para direita, mostram a condição "On" ou "Off".

Essas condições correspondem aos valores booleanos TRUE e FALSE. Uma variável booleana é vinculada a cada contato. Se essa variável é TRUE, então a condição é passada pela linha de conexão da esquerda para a direita; caso contrário, a conexão da direita recebe o valor "Off".

Contatos podem ser conectados em paralelo, então, uma das ramificações paralelas deve transmitir o valor "On" para que a ramificação paralela transmita o valor "On". Ou, os contatos são conectados em série então eles devem transmitir a condição "On" para que o último contato transmita a condição "On". Isso, portanto, corresponde a um “circuito elétrico” paralelo / série.

Um contato pode ser também negado, reconhecido pela barra no símbolo de contato: |/|. Então, o valor da linha é transmitido, se a variável for FALSE.

Bobina

No lado direito de uma rede em LD pode haver qualquer número das chamadas bobinas, que são representados por parênteses: ( ). Elas só podem ser paralelas. Uma bobina transmite o valor das conexões da esquerda para direita e copia em uma variável booleana apropriada. Na entrada da linha, o valor ON (corresponde à variável booleana TRUE) ou o valor OFF (correspondente à FALSE) pode estar presente.

Contatos e bobinas também podem ser negados (no exemplo o contato SWITCH1 e a bobina %QX3. 0 são negados). Se um bobina é negada (reconhecido pela barra no símbolo de bobina: (/)), então, ela copiará o valor negado na variável booleana apropriada. Se um contato é negado, em seguida, ele somente será conectado se o valor booleano apropriado for FALSE.

Blocos funcionais no Diagrama Ladder

Juntamente com contatos e bobinas, pode-se também entrar com blocos funcionais e programas. Na rede eles devem ter uma entrada e uma saída com valores booleanos e podem ser usados nos mesmos locais que os contatos – ou seja, no lado esquerdo da rede LD.

Bobinas Set/Reset

Bobinas podem ser definidas como bobinas Set e Reset. A bobina Set é caracterizada pelo “S”no símbolo das bobinas: (S). Ela nunca escreve sobre o valor TRUE em uma variável booleana apropriada. Isto é, se a variável foi uma vez definida como TRUE, ela permanecerá assim.

A bobina Reset é caracterizada pelo “R” no símbolo das bobinas: (R). Ela nunca escreve sobre o valor FALSE em uma variável booleana apropriada. Se a variável foi uma vez definida como FALSE, ela permanecerá assim.

LD como FBD

Ao trabalhar com LD é bem possível que se deseje usar o resultado do intertravamento de contatos para controlar outras POUs. Por outro lado, pode-se usar as bobinas para associar o resultado em uma variável global, a qual pode então ser usada em outro lugar. Pode-se, entretanto, também inserir uma possível chamada diretamente na sua rede LD. Para isso introduz-se uma POU com uma entrada EN.

Tais POUs são operandos completamente normais, funções, programas ou blocos funcionais que têm uma entrada adicional indicada com EN. A entrada EN é sempre do tipo BOOL e seu significado é: a entrada POU com EN é avaliada quando EN tem o valor TRUE.

Uma POU EN é ligada paralelamente às bobinas e a entrada EN é conectada à linha de conexão entre os contatos e as bobinas. Se as informações ON são transmitidas por esta linha, essa POU será avaliada normalmente.

Iniciando a partir de uma POU EN, pode-se criar redes semelhantes ao FBD.

Exemplo de uma rede LD com uma POU EN:

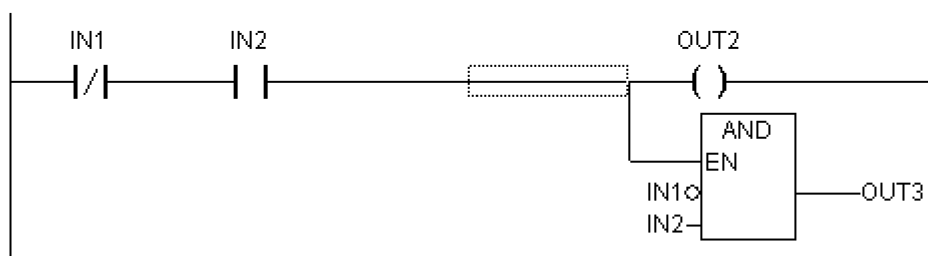


Figura 2-15. Exemplo de uma rede LD com uma POU EN

Depuração e funções online

Rastreamento

O Rastreamento de amostragem permite rastrear o valor da seqüência das variáveis, dependendo do assim chamado evento disparador, que consiste na borda de subida de uma variável booleana previamente definida (variável de disparo). O MasterTool IEC permite o rastreamento de até 20 variáveis. Para cada variável, 500 valores podem ser rastreados.

Depuração

As funções de depuração do MasterTool IEC facilitam a localização de erros.

Para depurar, execute o comando 'Projeto' 'Opções' e, na caixa de diálogo que aparece abaixo de Compilar, selecione ativar a opção Depurar.

Breakpoint

Um breakpoint é um local no programa onde o processamento é interrompido. Assim, é possível examinar os valores das variáveis em locais específicos dentro do programa.

Os breakpoints podem ser definidos em todos os editores. Nos editores textuais, breakpoints são definidos em números de linha, no FBD e LD em números de rede, no CFC em POUs e, em SFC, em passos. Nenhum breakpoint pode ser definido em instância de blocos funcionais.

Atenção: o controlador MasterTool IEC SP 32 Bit Full irá desativar a função cão de guarda da tarefa em questão assim que a execução do programa atual for interrompida em um breakpoint.

Passo a passo

Passo a passo significa:

Em IL: executar o programa até o próximo comando CAL, LD ou JMP.

Em ST: executar a próxima instrução.

Em FBD e LD: executar a próxima rede.

Em SFC: continuar a ação até o próximo passo.

Continuando passo a passo, pode-se verificar a lógica do seu programa.

Ciclado

Se Ciclado foi escolhido, a execução é interrompida após cada ciclo.

Alterar valores online

Durante as operações, variáveis podem ser definidas uma vez em um determinado valor (valor de escrita) ou também descritas novamente com um determinado valor após cada ciclo (valor forçado). No modo online, também é possível alterar o valor da variável por um clique duplo no valor. Deste modo, variáveis booleanas alternam-se entre TRUE e FALSE. Para cada um dos outros tipos de variáveis está disponível a janela‘Escrever variável xy’, onde o valor atual da variável pode ser editado.

Monitoração

No modo online, todas as variáveis exibíveis são lidas a partir o controlador e exibidas em tempo real. Esta exibição ocorre nas declarações e no editor de programa; pode-se também ler valores de variáveis atuais no gerenciador específico e em uma visualização. Se variáveis de instâncias de blocos funcionais devem ser monitoradas, a instância correspondente deve primeiro ser aberta.

Na monitoração das variáveis VAR_IN_OUT, o valor referenciado é o de saída.

Na monitoração de ponteiros, ambos, o ponteiro e o valor referência são considerados saídas na parte de declaração. Na parte do programa, somente o ponteiro é considerado saída:

+ --pointervar = '<'pointervalue'>'

PONTEIROS no valor referenciado também são exibidos adequadamente. Com um clique simples na cruz ou um clique duplo na linha, a exibição é expandida ou truncada.

Exemplo de monitoração de ponteiros:

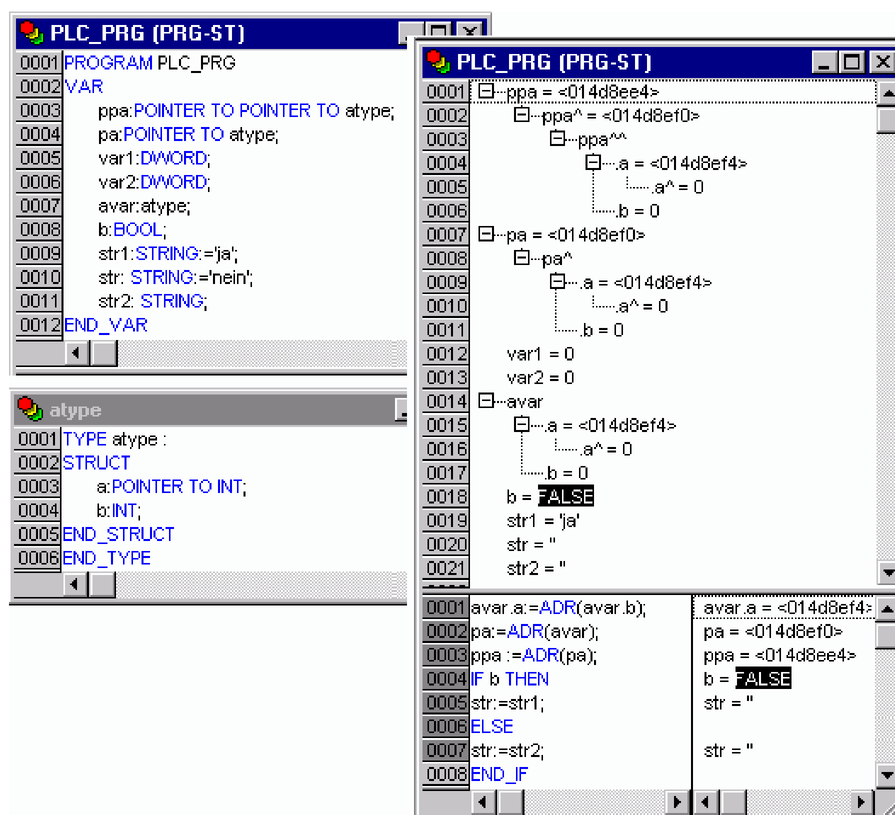


Figura 2-16. Exemplo de monitoração de ponteiros

Na implementação, o valor do ponteiro é exibido (veja também: de-referenciação). Para de-referenciação, entretanto, o valor de-referenciado é exibido.

Monitoração de matrizes: além de componentes de matrizes indexadas por uma constante, tais elementos também podem ser indexados por uma variável:

ANARRAY [1] = 5

ANARRAY [I] = 1

Se o índice consistir em uma expressão (ex.: [i j] ou [i 1]), o componente não pode ser exibido.

Considerar: se o número máximo de variáveis que podem ser monitoradas foi atingido, para cada variável adicional, em vez do valor atual, a mensagem "Muitas variáveis em monitoração" será exibida.

Simulação

Durante a simulação o programa criado para o CP não é processado no mesmo, mas sim no sistema “calculador” onde o MasterTool IEC está sendo executado. Todas as funções online estão disponíveis. Isto permite testar a lógica do programa sem o hardware do CP.

Considerar: POUs de bibliotecas externas não são executadas no modo de simulação.

A norma

A norma IEC 61131 3 é um padrão internacional para linguagens de programação de CPs.

As linguagens de programação oferecidas no MasterTool IEC estão de acordo com os requisitos da norma.

De acordo com a essa norma, um programa consiste nos seguintes elementos:

- Estruturas (consultar tipos de dados)
- POUs
- Variáveis globais

Os elementos de linguagem gerais são descritos nas seções: Identificador, Endereços, Tipos, Comentários e Constantes.

O processamento de um programa MasterTool IEC começa com a POU especial PLC_PRG. A POU PLC_PRG pode chamar outras POUs.

3. Edição de um programa no MasterTool IEC

Controlando uma unidade de sinal de trânsito

Vamos agora começar a escrever um exemplo de um pequeno programa. Trata-se de uma simples unidade de sinal de trânsito que deve controlar dois sinais de trânsito em uma intersecção. A fase vermelha/verde de ambos os sinais de trânsito são alternadas e, para prevenir acidentes, nós vamos inserir fases de transição amarela ou amarelo-vermelha. As últimas serão mais longas que as primeiras.

Neste exemplo, ilustra-se quantos programas dependentes do tempo podem ser mostrados com os recursos de linguagem da norma IEC´61131-3, como é possível editar diferentes linguagens do padrão com a ajuda do **MasterTool IEC** e também como pode-se facilmente conectá-los enquanto se familiariza com a simulação do **MasterTool IEC.**

Criação de uma POU

Iniciar sempre é fácil: abra o **MasterTool IEC** e selecione 'Arquivo' 'Novo'.

Na caixa de diálogo que aparece, percebe-se que já foi dado um nome padrão a primeira POU: PLC_PRG. Mantenha esse nome e o tipo da POU deve definitivamente ser um programa. Nesse caso, nós escolhemos o editor Gráfico Contínuo de Funções (CFC) para ser a linguagem desta POU.

Agora crie mais três objetos com o comando 'Projeto' 'Acrescentar Objeto', na barra do menu ou no menu de contexto (pressione o botão direito do mouse no Organizador de objetos): um programa na linguagem Seqüenciamento Gráfico de Funções (SFC) nomeado SEQUENCE, um bloco funcional na linguagem Diagrama de Blocos Funcionais (FBD) nomeado TRAFFICSIGNAL, juntamente com a POU WAIT, também do tipo bloco funcional, a qual nós desejamos programar como Lista de Instruções (IL).

O que a POU TRAFFICSIGNAL faz?

Na POU TRAFFICSIGNAL nós vamos atribuir fases individuais às luzes, isto é, nós vamos garantir que a luz vermelha fique vermelha na fase vermelha e na fase amarelo-vermelha, a luz amarela nas fases amarela e amarelo-vermelha, etc.

O que a POU WAIT faz?

Em WAIT nós vamos atribuir um simples temporizador que, como entrada, receberá o intervalo de tempo da fase em milissegundos e, como saída, produzirá TRUE assim que o período de tempo tiver terminado.

O que a POU SEQUENCE faz?

Em SEQUENCE tudo é combinado de forma que a luz correta ilumine pelo tempo correto no período desejado.

O que a POU PLC_PRG faz?

No PLC_PRG a entrada do sinal de início está conectada a seqüência de luzes de trânsito e as “instruções de cor”, para cada lâmpada são providenciadas como saídas.

Simulação preliminar do TRAFFICSIGNAL

Agora teste seu programa no modo de simulação. Compile (**'Projeto' 'Compilar'**) e carregue-o (**'Comunicação' 'Login'**). Inicie o programa com '**Comunicação**' '**Run**', então ajuste a variável ON para TRUE, por exemplo, com um clique duplo na entrada ‘ON‘ na caixa de entrada do editor CFC. Isto deixará a variável preparada para ser ajustada para <TRUE>. Então pressione <Ctrl> + <F7> ou o comando 'Comunicação' 'Escrever valores' para ajustar o valor. Agora a variável START em SEQUENCE (que nós ajustamos para TRUE manualmente no primeiro nível do programa) fica com o valor pela variável ON, que é usada no PLC_PRG. Isto fará rodar os ciclos de luzes. PLC_PRG muda então para a janela de monitoração. Clique duas vezes no sinal de mais no editor de declaração, a variável abre, e pode-se ver os valores das variáveis individuais.

Declaração do TRAFFICSIGNAL

Vamos agora analisar a POU TRAFFICSIGNAL. No editor de declaração declara-se como variável de entrada (entre as palavras-chave VAR_INPUT e END_VAR) a variável nomeada STATUS do tipo INT. O STATUS terá quatro condições possíveis, uma para cada fase do TRAFFICSIGNAL: verde, amarela, amarelo-vermelha e vermelha.

Além disso, o nosso TRAFFICSIGNAL tem três saídas, que são RED (vermelha), YELLOW (amarela) e GREEN (verde). Deve-se declarar estas três variáveis. Então a parte da declaração do bloco funcional TRAFFICSIGNAL se parecerá assim:

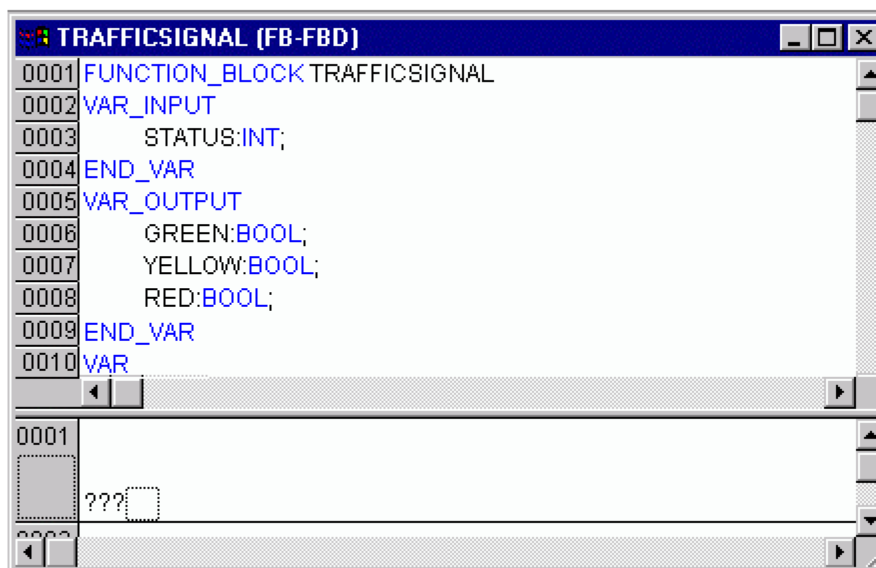
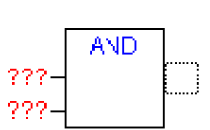


Figura 3-1. Declaração do bloco funcional TRAFFICSIGNAL

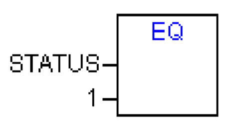
Corpo do TRAFFICSIGNAL

Agora nós determinaremos os valores das variáveis de saída dependendo da entrada STATUS da POU. Para fazer isso, vá ao corpo da POU. Clique no campo da esquerda ao lado da primeira rede (o campo cinza com o número 0001). Está selecionada agora a primeira rede. Deve-se escolher o item do menu 'Inserir' 'Caixa (Box)'.

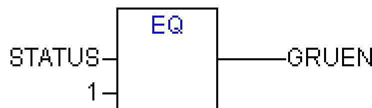
Na primeira rede uma caixa é inserida com o operador AND e duas entradas:



Clique no texto AND, então ele aparece selecionado e muda texto para EQ. Selecione então, para cada uma das duas entradas, os três pontos de interrogação e sobrescreva-os com "STATUS" e "1", respectivamente.



Clique agora no lado à direita da caixa EQ. Assim, a saída da operação EQ está selecionada. Escolha 'Inserir' 'Atribuição'. Altere os três pontos de interrogação “???” para GREEN. Foi criada uma rede com a seguinte estrutura.



STATUS é comparado com 1, o resultado é atribuído a GREEN. Esta rede, desta forma, altera-se para GREEN, se o valor do estado de preset é 1.

Para as outras cores do TRAFFICSIGNAL nós precisamos de mais duas redes. Para criar a primeira, execute o comando **'Inserir' 'Rede (depois)'** e insira uma caixa EQ conforme descrito acima. Selecione então o pino de saída desta caixa e use novamente o comando **'Inserir' 'Caixa (Box)'**. Na nova caixa substitua “AND” por ”OR”. Agora selecione o primeiro pino de saída da caixa OR e use o comando **'Inserir' 'Atribuição'** para atribuí-lo à próxima cor. Selecione a segunda entrada da caixa OR por um clique no mouse na linha horizontal próxima aos três pontos de interrogação, então ela aparece marcada por um retângulo pontilhado. Agora use **'Inserir' 'Caixa (Box)'** para adicionar uma caixa EQ adicional como descrito acima. Finalmente a rede deve parecer-se com o que está mostrado a seguir:

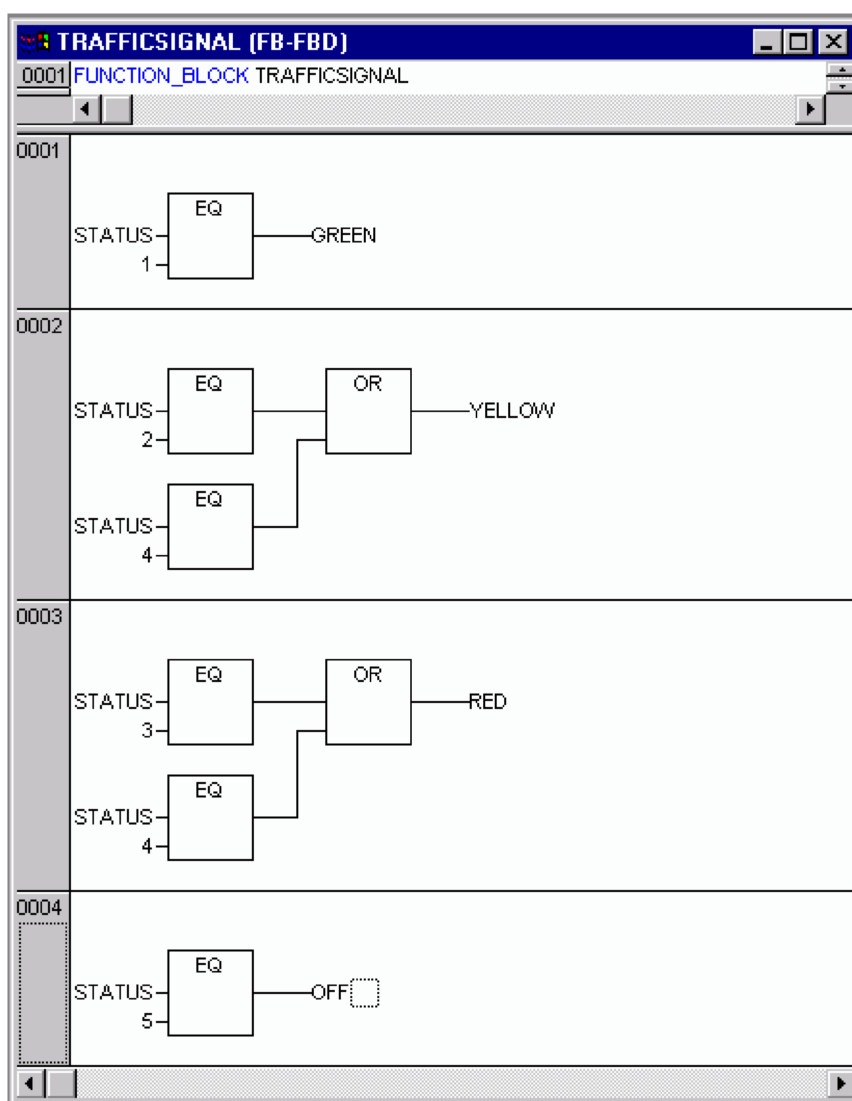


Figura 3-2. Bloco funcional TRAFFICSIGNAL, parte de instruções

Para inserir um operador em frente a outro operador, deve-se selecionar o local na caixa (box) onde deseja-se anexar o novo operador.

Então use o comando 'Inserir' 'Caixa (Box)'. Caso contrário, não será possível configurar estas redes da mesma maneira que a primeira rede.

Agora nossa primeira POU foi concluída. TRAFFICSIGNAL, de acordo com o valor da entrada STATUS, controla qualquer cor de luz que desejarmos.

Conectando a biblioteca padrão

Para o temporizador na POU WAIT, nós precisamos de uma POU da biblioteca padrão. Portanto, abra o Gerenciador de Bibliotecas com **'Janela' 'Gerenciador de Bibliotecas'**. Escolha 'Inserir' 'Biblioteca adicional'. A caixa de diálogo aparece para abertura de arquivos. A partir da lista de bibliotecas, escolha standard.lib.

Declaração do WAIT

Agora vamos nos concentrar na POU WAIT. Esta POU deve tornar-se um temporizador no qual nós podemos determinar o período de tempo de cada fase do TRAFFICSIGNAL. Nossa POU recebe como variável de entrada, a variável TIME_IN do tipo TIME, e como saída ela produz um valor booleano, o qual nós queremos chamar de OK e que deve ser TRUE quando o período de tempo

desejado transcorrer. Nós definimos este valor como FALSE inserindo, ao final da declaração e antes do ponto e vírgula, a atribuição “:=FALSE”.

Para tanto, nós precisamos da POU TP: um gerador de sinais de relógio (clock). Este tem duas entradas (IN, PT) e duas saídas (Q, ET). TP faz o seguinte:

Enquanto IN for FALSE, ET será 0 (zero) e Q será FALSE. Assim que IN fornecer o valor TRUE, o tempo é calculado na saída ET em milissegundos. Quando ET alcançar o valor PT, então ET não será mais incrementado. Enquanto isto, Q produz TRUE enquanto ET for menor que PT. Assim que o valor PT for alcançado, então Q produzirá FALSE novamente. Consultar o capítulo para descrições breves de todas as POUs da biblioteca padrão.

Para usar a POU TP na POU WAIT, deve-se criar uma instância local a partir de TP. Para isto declara-se uma variável local ZAB (tempo transcorrido) do tipo TP (entre as palavras-chave VAR, END_VAR).

A parte das declarações do WAIT se parece com o que segue:

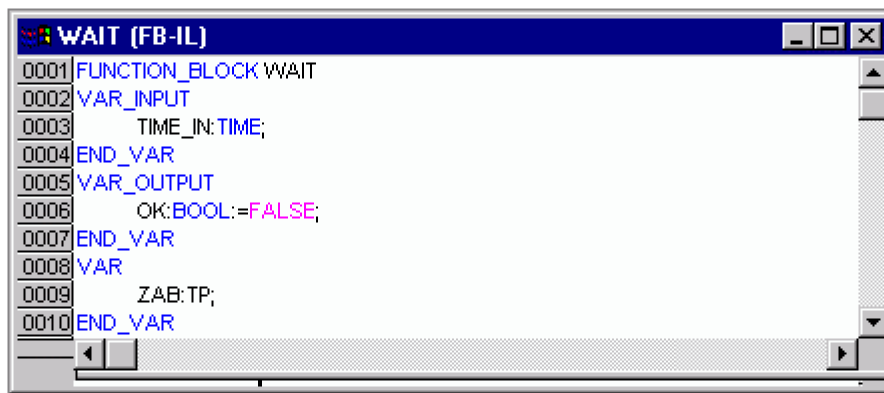


Figura 3-3. Bloco funcional WAIT, parte declarações

Corpo do WAIT

Para criar o timer desejado, o corpo da POU deve ser programado como a seguir:

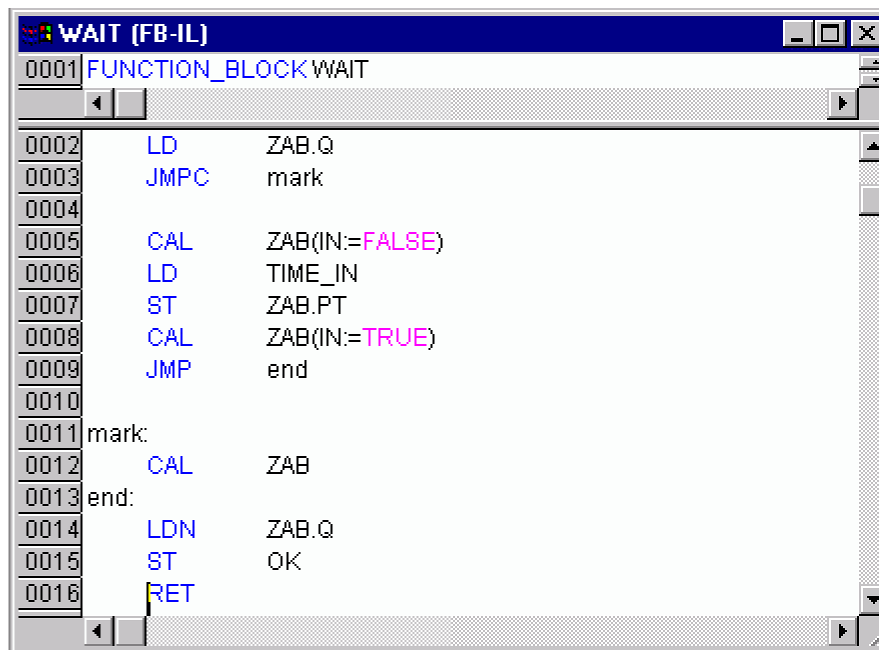


Figura 3-4. Bloco funcional WAIT, parte de instruções

Primeiramente é verificado se Q já foi definido para TRUE (a contagem já foi executada), nesse caso, nós não alteramos nada com a ocupação de ZAB, mas nós chamamos o bloco funcional ZAB sem entrada (para verificar se o período de tempo já transcorreu).

Caso contrário, nós definimos a variável IN em ZAB como FALSE e, portanto, ao mesmo tempo ET como 0 (zero) e Q como FALSE. Desse modo, todas as variáveis são configuradas na condição inicial desejada. Agora nós atribuímos o tempo necessário a partir da variável TIME na variável PT, e chamamos ZAB com IN:=TRUE. No bloco funcional ZAB, a variável ET é agora calculada até alcançar o valor TIME, então Q é definido como FALSE.

O valor negado de Q está em OK após cada execução de WAIT. Assim que Q assumir o estado FALSE, então OK produz TRUE.

O temporizador é concluído neste ponto. Agora é hora de combinar nossos dois blocos funcionais WAIT e SEQUENCE no programa principal PLC_PRG.

Primeiro nível de expansão do SEQUENCE

Primeiro nós declaramos as variáveis que precisamos. São elas: uma variável de entrada START do tipo BOOL, duas variáveis de saída TRAFFICSIGNAL1 e TRAFFICSIGNAL2 do tipo INT e uma do tipo WAIT (DELAY). O programa SEQUENCE agora se parece com o mostrado a seguir:

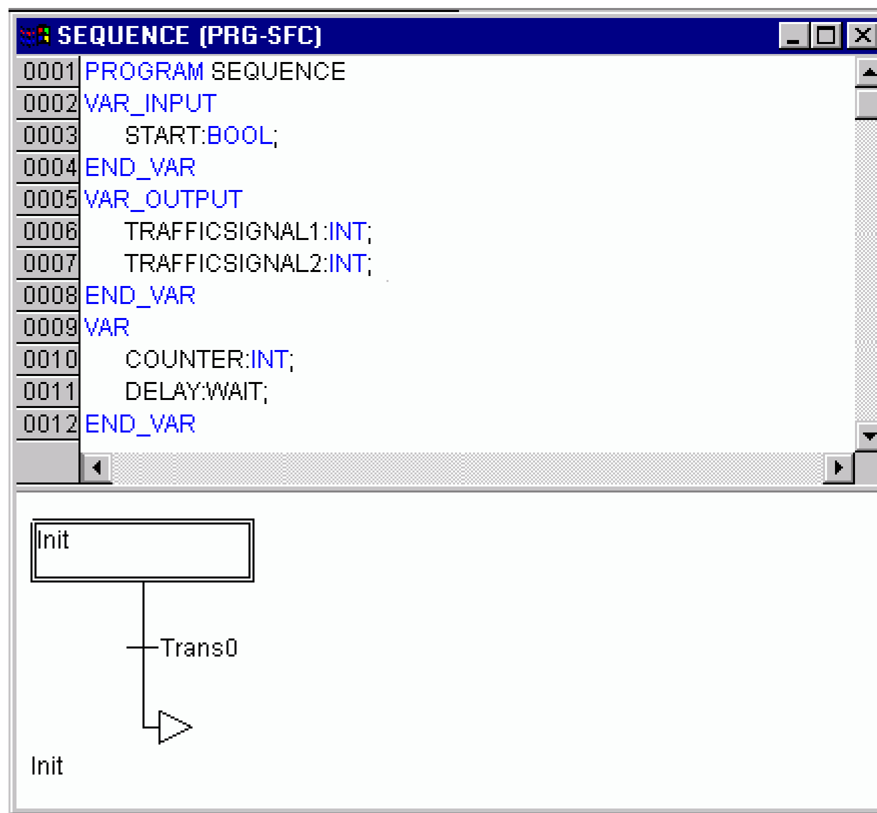


Figura 3-5. Programa SEQUENCE, Primeiro Nível de Expansão, Parte Declarações

Criação de um diagrama SFC

O diagrama inicial de uma POU no SFC sempre consiste de uma ação "Init", de uma transição associada "Trans0" e de um salto de volta a Init. Nós precisamos expandir isto.

Antes de programar a ação individual e as transições vamos primeiro determinar a estrutura dos diagramas. Nós precisamos de uma etapa para cada fase TRAFFICSIGNAL. Insira-a marcando Trans0 e escolhendo 'Inserir' 'Transição de passo (depois)'. Repita este procedimento mais três vezes.

Ao clicar diretamente no nome de uma transição ou de um passo, então este está marcado e pode-se mudá-lo. Nomeie a primeira transição após Init como “START”, e todas as outras transições “DELAY. OK”.

A primeira transição ocorre quando START é TRUE e todas outras ocorrem quando DELAY em OK produz TRUE, isto é, quando o período de tempo configurado foi finalizado.

Os passos (de cima a baixo) recebem os nomes Switch1, Green2, Switch2, Green1, enquanto Init naturalmente mantém seu nome. "Switch" deve incluir uma fase amarela, no Green1 TRAFFICSIGNAL1 será verde, no Green2 TRAFFICSIGNAL2 será verde. Finalmente altere o endereço de retorno de Init depois de Switch1. Se tudo estiver correto, então o diagrama deve se parecer com o seguinte:

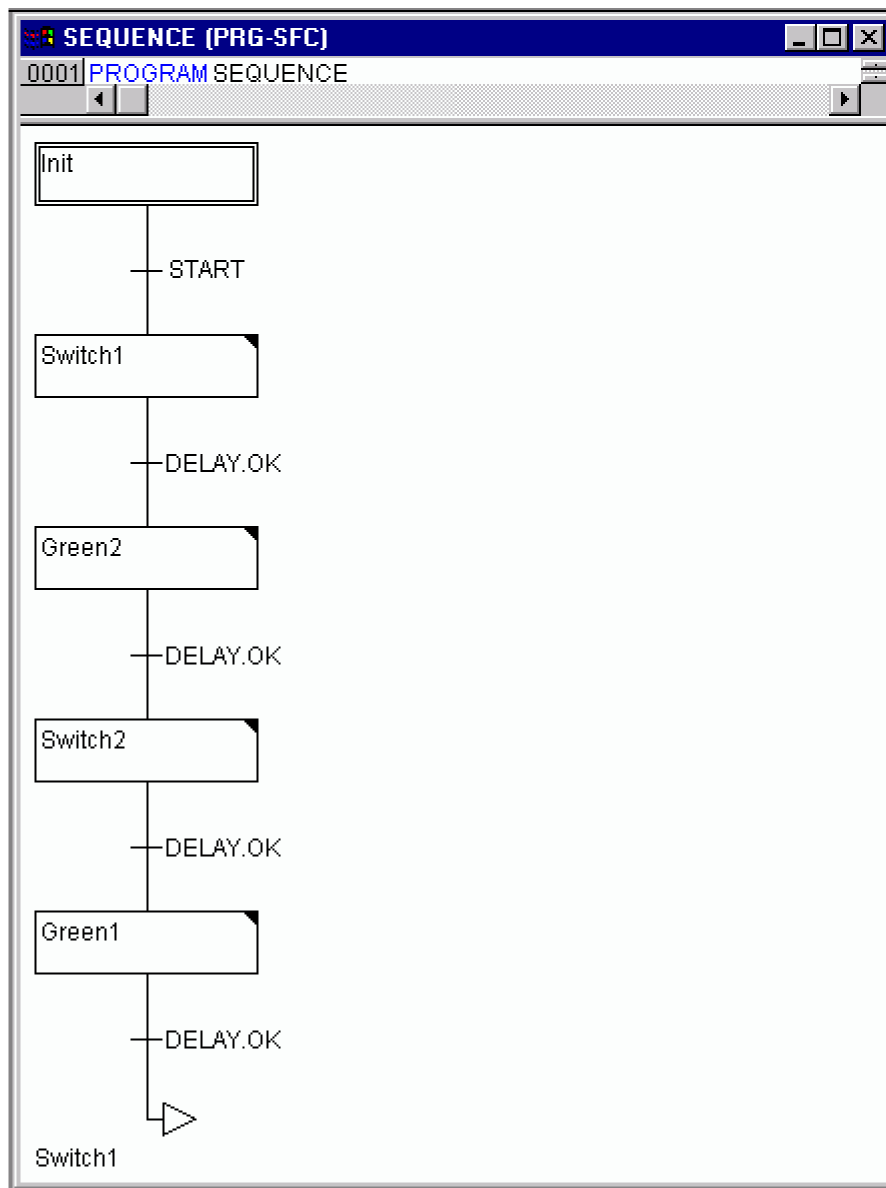


Figura 3-6. Programa SEQUENCE, Primeiro Nível de Expansão, Parte Instruções

Agora finaliza-se a programação dos passos individuais. Com um clique duplo no campo de um passo, então obtém-se um diálogo para abrir uma nova ação. Nesse caso, será usada IL (Lista de Instruções).

Ações e condições de transições

Na ação do passo Init as variáveis são inicializadas, o STATUS do TRAFFICSIGNAL1 deve ser 1 (verde). O estado do TRAFFICSIGNAL2 deve ser 3 (vermelho). A ação Init então se parece com a imagem seguinte:

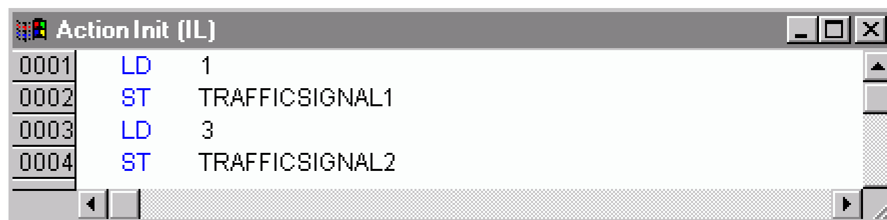


Figura 3-7. Ação Init

Switch1 muda o estado do TRAFFICSIGNAL1 para 2 (amarelo), e o do TRAFFICSIGNAL2 para 4 (amarelo-vermelho). Adicionalmente, um retardo de 2000 milissegundos é configurado. A ação é agora como segue:

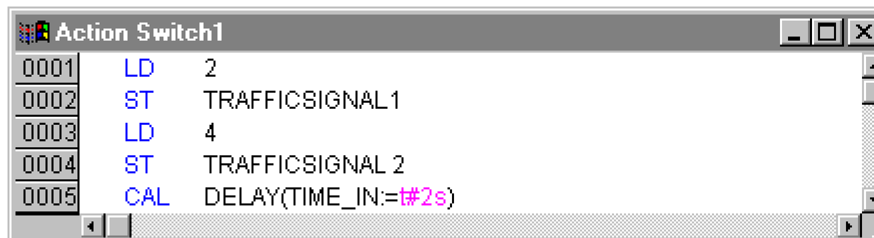


Figura 3-8. Ação Switch1

Com Green2 TRAFFICSIGNAL1 está vermelho (STATUS:=3), TRAFFICSIGNAL2 verde (STATUS:=1) e o tempo de retardo é de 5000 milissegundos.

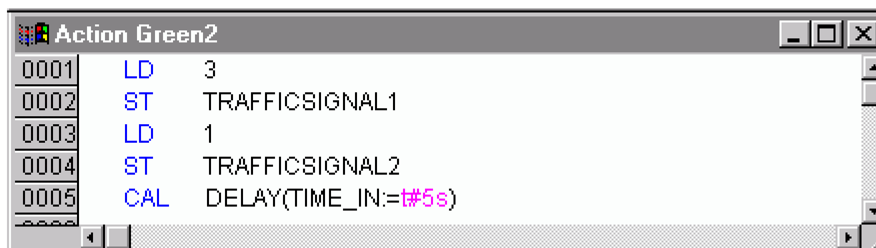


Figura 3-9. Ação Green2

No Switch2 o STATUS de TRAFFICSIGNAL1 altera para 4 (amarelo-vermelho), o do TRAFFICSIGNAL2 para 2 (amarelo). Um tempo de retardo de 2000 milissegundos está agora configurado.

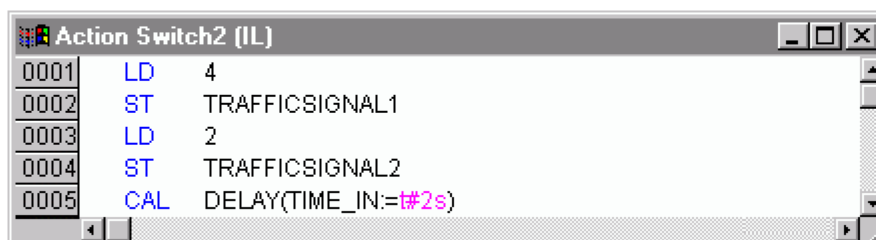


Figura 3-10. Ação Switch2

Com Green1 TRAFFICSIGNAL1 é verde (STATUS:=1), TRAFFICSIGNAL2 é vermelho (STATUS:=3) e o tempo de retardo é configurado para 5000 milissegundos.

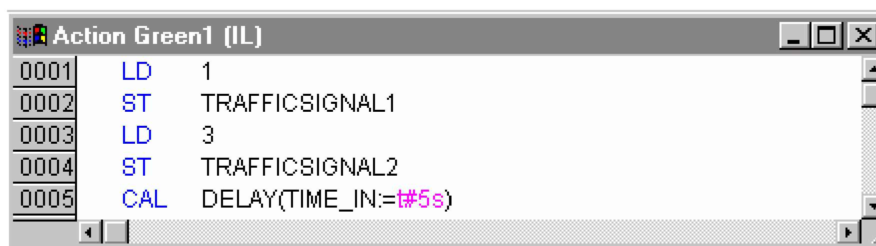


Figura 3-11. Ação Green1

A primeira fase de expansão do nosso programa está concluída.

Para fazer um primeiro teste da POU SEQUENCE no modo de simulação, deve-se executar as seguintes etapas:

1. Abrir a POU PLC_PRG. Cada projeto inicia executando com PLC_PRG. Para ser possível iniciar a POU SEQUENCE, inserir uma caixa e substituir “AND” por “SEQUENCE”. As entradas e saídas devem ser mantidas sem atribuição neste momento.

2. Compilar o projeto via 'Projeto' 'Compilar'. A janela de mensagens deve indicar “0 Erros, 0 Advertências”. Agora deve-se verificar se a opção 'Comunicação' 'Modo simulação' está ativada e usar o comando 'Comunicação' 'Login' para entrar no modo de simulação. O programa deve ser iniciado com 'Comunicação' 'Run'. A POU SEQUENCE deve ser aberta com um clique duplo na entrada “SEQUENCE” no Organizador de objetos. O programa agora está iniciado, mas para colocá-lo em execução, a variável START deve ser TRUE. Mais tarde esta será configurada por PLC_PRG, mas, por enquanto, é necessário configurá-la manualmente dentro da POU. Para fazer isto, deve-se efetuar um clique duplo na linha na parte declarações, onde START está definida (START:=FALSE). Isto irá definir a opção "<:=TRUE>" atrás da variável na cor turquesa. Agora seleciona-se o comando 'Comunicação' 'Escrever valores' para configurar este valor. A seguir START será indicado por uma marca azul do passo ativo atual.

Ao concluir esse teste intermediário usa-se o comando 'Comunicação' 'Logout' para sair do modo de simulação e continuar a programação.

Segundo nível de expansão do SEQUENCE

Para garantir que o nosso diagrama tenha pelo menos uma ramificação alternativa e, para que assim nós possamos desligar nosso sinal de trânsito à noite, nós agora vamos incluir em nosso programa um contador que, depois de certo número de ciclos de TRAFFICSIGNAL, desliga a unidade.

Primeiramente nós precisamos de uma nova variável COUNTER do tipo INT. Declare-a como normalmente se faz na parte declarações da SEQUENCE, e inicialize-a em Init com 0 (zero), como na figura abaixo.

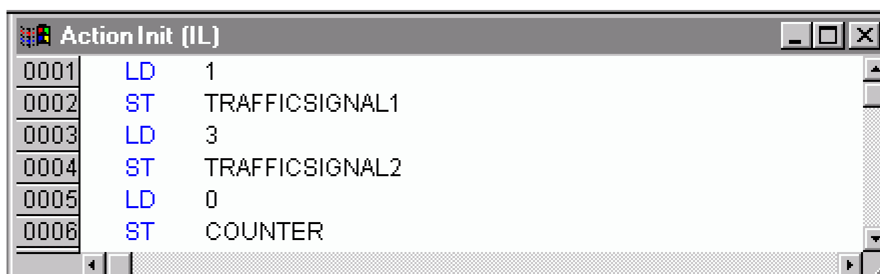


Figura 3-12. Ação Init, Segunda Versão

Agora selecione a transição depois de Switch1 e insira um passo e depois a transição. Selecione a transição resultante e insira uma ramificação alternativa à sua esquerda. Após esta transição à esquerda insira um passo e uma transição. Depois da nova transição resultante, insira um salto após Switch1.

Nomeie as novas partes como segue: a superior dos novos dois passos deve ser chamada “Count” e a inferior “Off”. As transições são chamadas (de cima a baixo e da esquerda para direita) EXIT, TRUE e DELAY. OK. A nova parte deve parecer-se com a parte marcada com a borda preta na imagem seguinte:

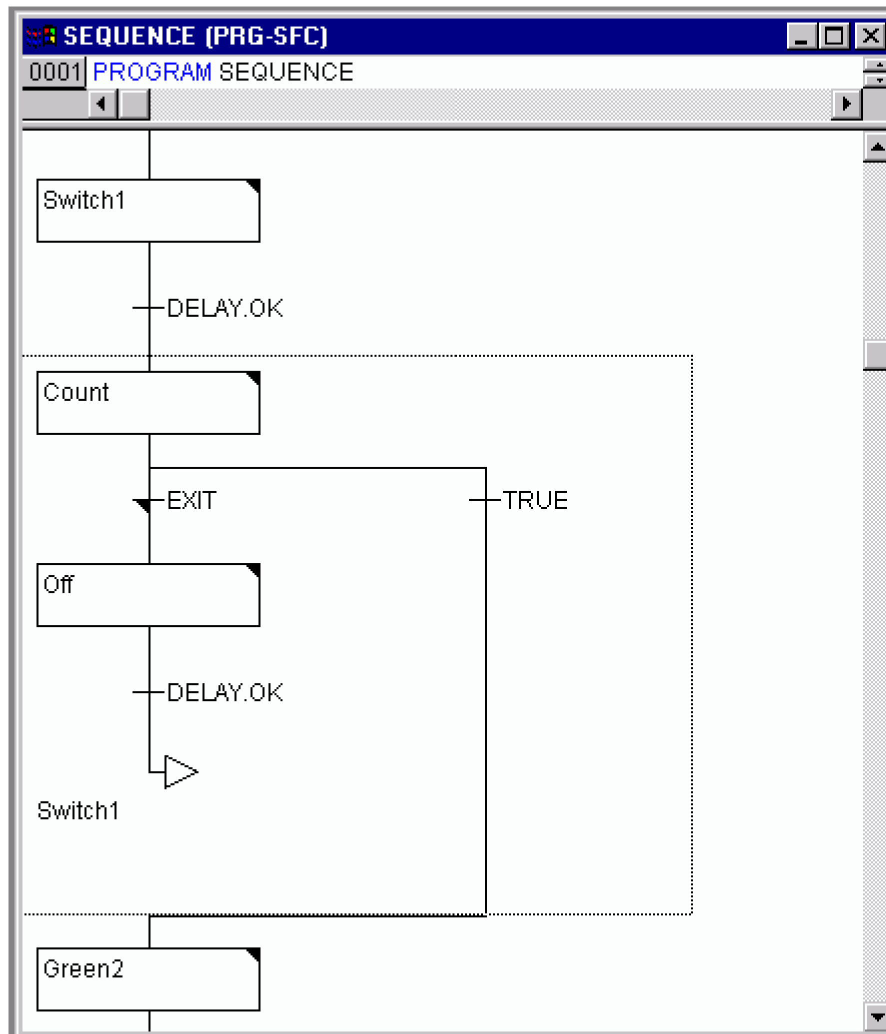


Figura 3-13. Programa SEQUENCE, Segundo nível de expansão, Parte Instruções

Agora duas novas ações e uma nova condição de transição devem ser implementadas. Na etapa Count, a variável COUNTER é incrementada em 1:

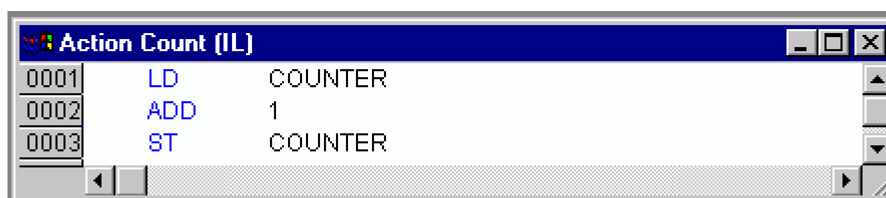


Figura 3-14. Ação Count

A transição EXIT verifica se o contador é maior que um determinado valor, por exemplo, 7:



Figura 3-15. Transição EXIT

Em Off, o estado de ambas as luzes estão definidos em 5(OFF) (ou qualquer outro número diferente de 1, 2, 3 ou 4), COUNTER é levado para 0 (zero) e um tempo de retardo de 10 segundos é configurado:

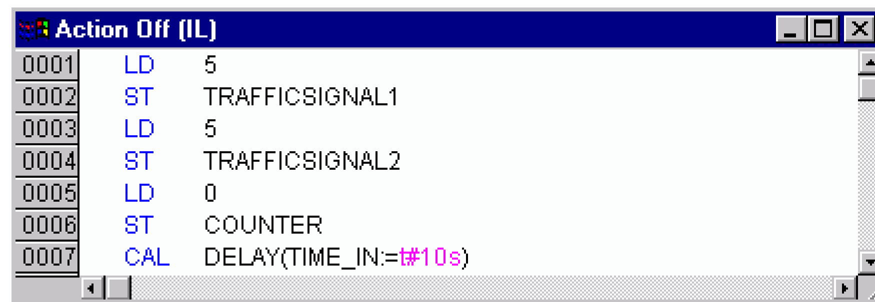


Figura 3-16. Ação Off

O resultado

Em nossa situação hipotética, cai a noite depois de sete ciclos do sinal de trânsito. Por 10 segundos as sinaleiras desligam-se e então nós temos a luz do dia novamente; a unidade de luz de trânsito liga-se de novo e todo o processo começa novamente do início. Se desejado, pode-se fazer outro teste antes da versão atual do seu programa no modo simulação antes que nós continuemos a criar a POU PLC_PRG.

PLC_PRG

Nós definimos e correlacionamos o seqüenciamento de tempo das fases para ambas as unidades de sinais de trânsito no bloco SEQUENCE. Uma vez que nós encaramos o sistema de sinal de trânsito como um módulo de sistema de barramento (barramento CAN, por exemplo), nós temos que tornar disponíveis as variáveis de entrada e saída no bloco PLC_PRG. Nós queremos iniciar o sistema de sinal de trânsito em cima de ON e nós queremos enviar cada uma das seis lâmpadas (cada sinal de trânsito: vermelha, verde e amarela) e o correspondente “comando de sinal” para cada etapa da SEQUENCE. Nós agora estamos declarando variáveis booleanas apropriadas para estas seis saídas e uma entrada (antes de criarmos o programa no editor) e estamos alocando-as, ao mesmo tempo, no correspondente endereço IEC.

O próximo passo é declarar as variáveis Light1 e Light2 do tipo TRAFFICSIGNAL na declaração do editor (veja a figura).

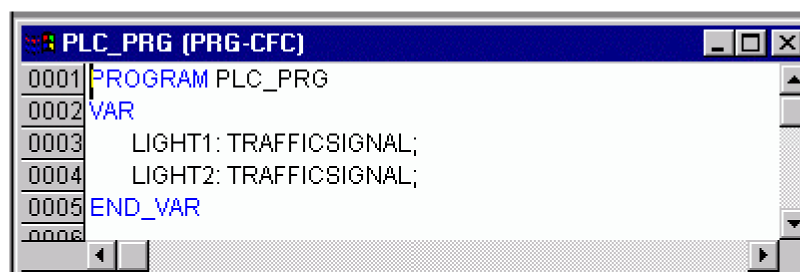


Figura 3-17. Declarações LIGHT1 e LIGHT2

Estas entregam o valor booleano de cada uma das seis luzes às seis saídas mencionadas acima, para cada passo do bloco SEQUENCE. Entretanto, nós não estamos declarando as variáveis de saída que foram vistas dentro do bloco PLC_PRG e sim aquelas nos Recursos para Variáveis Globais. A

variável de entrada booleana IN, que é usada para configurar a variável START no bloco SEQUENCE para TRUE, pode ser configurada da mesma maneira. ON é também alocada para um endereço IEC.

Selecione a guia Recursos e abra a lista de Variáveis globais.

Faça a declaração como segue:

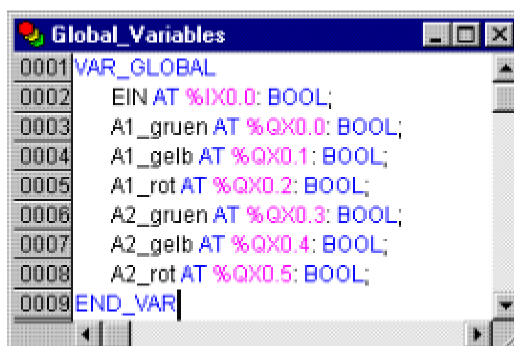


Figura 3-18. Declaração das variáveis de entrada/saída

O nome da variável (por exemplo, IN) é seguido, após AT, por um sinal PERCENTUAL, o qual indica o endereço IEC. I para entrada, Q para saída, B para byte e os bits individuais do módulo são endereçados usando 0.0 (0.1, 0.2, etc.). Não será feita a configuração do controlador necessária nesse exemplo porque ele depende de qual pacote do dispositivo está disponível no computador. Ver o item configurações do CP para maiores informações.

Para finalizar o bloco PLC_PRG, recorre-se à janela do editor. Seleciona-se o Editor CFC e conseqüentemente se obtém, abaixo da barra do menu, uma barra de símbolos CFC de todos os elementos disponíveis (veja Editor CFC)

Agora deve-se clicar com o botão direito do mouse na janela do editor e selecionar o elemento Caixa (Box). O texto AND deve ser substituído por "SEQUENCE". Isto traz o bloco SEQUENCE, com todas as variáveis de entrada e saída definidas. Insira dois elementos de bloco adicionais, aos quais dá-se o nome de TRAFFICSIGNAL. É um bloco funcional e faz com que se obtenha três pontos de interrogação vermelhos marcados em cima do bloco que devem ser substituídos com as já declaradas variáveis locais LIGHT1 e LIGHT2. Agora configura-se um elemento do tipo **Input**, ao qual é conferido o título ON e seis elementos do tipo **Output**, aos quais se confere nomes variáveis para, como descrito, L1_green, L1_yellow, L1_red, L2_green, L2_yellow, L2_red.

Todos os elementos do programa estão agora no lugar e pode-se conectar entradas e saídas clicando na linha curta na entrada/saída de um elemento, arrastando-o e mantendo o botão do mouse pressionado até a entrada/saída do elemento desejado.

O programa deve finalmente parecer-se com o exemplo mostrado a seguir:

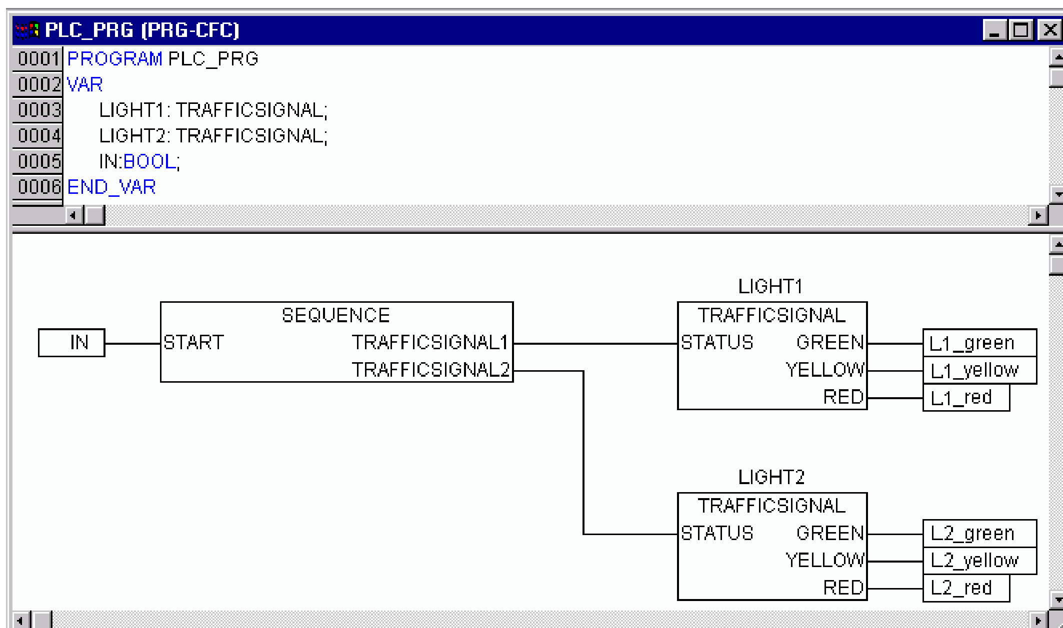


Figura 3-19. PLC_PRG, Declarações e apresentação com o editor CFC

Simulação do TRAFFICSIGNAL

Agora teste seu programa no modo de simulação. Compile ('**Projeto**' '**Compilar**') e carregue-o ('**Comunicação**' '**Login**'). Inicie o programa com **'Comunicação' 'Run'**, então ajuste a variável ON para TRUE, por exemplo, com um clique duplo na entrada 'ON' na caixa de entrada para o editor CFC. Isto deixará a variável preparada para ser ajustada para <TRUE>. Então pressione <Ctrl> + <F7> ou o comando 'Comunicação' 'Escrever valores' para ajustar o valor. Agora a variável START em SEQUENCE (que nós ajustamos para TRUE manualmente no primeiro nível do programa) fica com o valor pela variável ON, que é usada no PLC_PRG. Isso fará rodar os ciclos de luzes. PLC_PRG muda então para janela de monitoração. Clique duas vezes no sinal de mais no editor de declaração, a variável abre, e pode-se ver os valores das variáveis individuais.

Visualizando a unidade de sinal de trânsito

Com a visualização do **MasterTool IEC** pode-se fácil e rapidamente dar vida às variáveis do projeto. Vamos desenhar dois sinais de trânsito e uma chave ON para o sistema que ilustrarão o processo de chaveamento.

Criando uma nova visualização

Para criar uma visualização deve-se primeiro selecionar a aba da **Visualização** no Organizador de objetos. Primeiro deve-se clicar na borda inferior da janela, no lado esquerdo, no cartão de registro com o símbolo e o nome **Visualização**. Ao escolher o comando 'Projeto' 'Acrescentar objeto', uma caixa de diálogo será aberta.



Figura 3-20. Diálogo para criar uma visualização

Pode-se digitar aqui qualquer nome. Ao confirmar o diálogo com OK, uma janela abre, na qual pode-se configurar a nova visualização.

Inserindo elementos na tela

Para a visualização do sinal de trânsito, deve-se proceder como segue:

Executa-se o comando 'Inserir' 'Elipse'. Desenha-se um círculo de tamanho médio (~2 cm). Para isto, deve-se clicar no campo de edição e desenhar o círculo em seu comprimento com o botão esquerdo do mouse pressionado.

Agora via clique duplo no círculo a caixa de diálogo para edição de elementos de visualização abre.

Escolhe-se a categoria Variáveis e no campo Alterar Cor deve-se indicar a variável de nome. L1_red ou L1_red. Isto significa que a variável global L1_red causará a mudança de cor assim que for configurada para TRUE. O ponto antes do nome da variável indica que ela é uma variável global, mas isto não é mandatório.

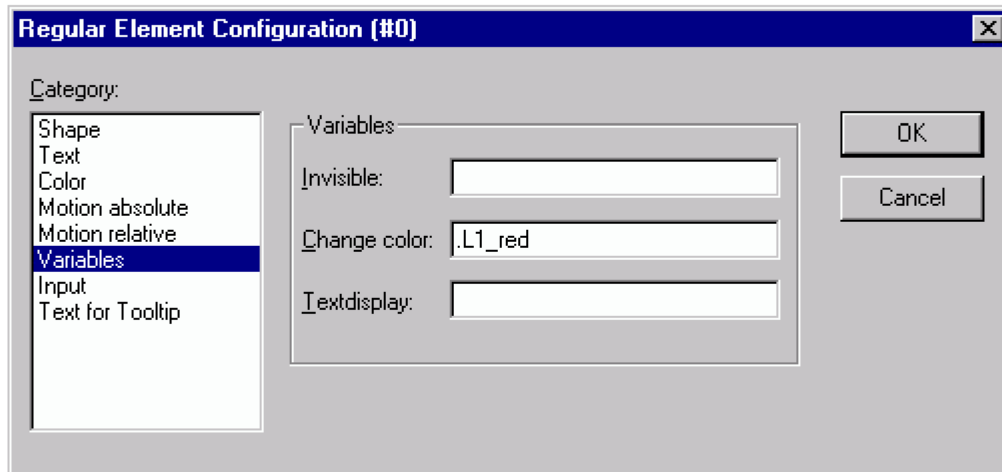


Figura 3-21. Caixa de diálogo para edição de elementos de tela

Então escolhe-se a categoria **Cor** e clica-se no botão **Dentro** na área **Cor**. Escolhe-se a cor mais neutra possível, como preto, por exemplo.

Deve-se clicar no botão **Dentro** na área **Cor do Alarme** e escolher a tonalidade de cor que mais se aproxima daquele da luz vermelha.

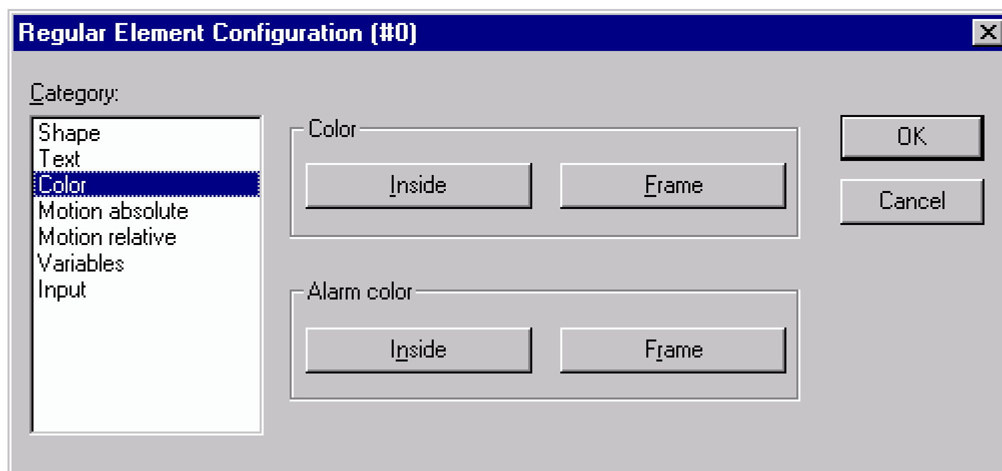


Figura 3-22. Ajuste da cor para o item

O círculo resultante normalmente será preto e , quando a variável L1_red do TRAFFICSIGNAL1 for TRUE, então a cor será alterada para vermelho. Portanto, criou-se a primeira luz do primeiro sinal!

As outras "luzes de trânsito"

Agora deve-se entrar com o comando 'Editar' 'Copiar' (<Ctrl>+<C>) e depois duas vezes 'Editar' 'Colar' (<Ctrl>+<V>). Isto gera dois círculos a mais, do mesmo tamanho, posicionados acima do primeiro. Pode-se mover os círculos clicando em um e arrastando-o (enquanto mantém o botão do mouse pressionado). A posição desejada deve, no nosso caso, estar em uma coluna vertical na metade esquerda da janela de edição. Deve-se dar um clique duplo em um dos dois círculos para abrir a caixa de diálogo de configuração novamente. Digita-se as seguintes variáveis no campo 'Alterar cor' do círculo correspondente:

- Para o círculo do meio: L1_yellow
- Para o círculo inferior: L1_green

Agora escolhe-se para os círculos, na categoria **Cor** e na área **Cor do alarme,** a cor correspondente (amarelo ou verde).

Finalizando o sinal de trânsito primário

Agora seleciona-se o comando 'Inserir' 'Retângulo' para inserir, do mesmo modo que o círculo, um retângulo que inclua os três círculos. Novamente escolhe-se uma cor, o mais neutra possível, para o retângulo e executa-se o comando **'Extras' 'Enviar para trás'**, para que os círculos tornem-se visíveis novamente.

Se o modo simulação ainda não está ligado, pode-se ativá-lo com o comando 'Comunicação' 'Modo simulação'.

Para iniciar a simulação utiliza-se os comandos 'Comunicação' 'Login' e 'Comunicação' 'Run', então pode-se observar a alteração de cor do primeiro sinal de trânsito.

O sinal de trânsito secundário

O caminho mais fácil para criar o segundo sinal de trânsito é copiar todos os elementos do primeiro. Para isto, seleciona-se todos os elementos do sinal de trânsito e os copia (como foi feito anteriormente com as cores do primeiro) com os comandos 'Editar' 'Copiar' e 'Editar' 'Colar'. Então só é necessário alterar o texto "TRAFFICSIGNAL1" nas respectivas caixas de diálogos para "TRAFFICSIGNAL2" e a visualização do segundo está concluída.

A chave ON

Insere-se um retângulo e confere-se a ele, conforme descrito acima, uma cor escolhida e escolhe-se ON nas **Variáveis** para **Mudança de Cor**. Digita-se "ON" no campo de entrada para **Conteúdo** na categoria Texto.

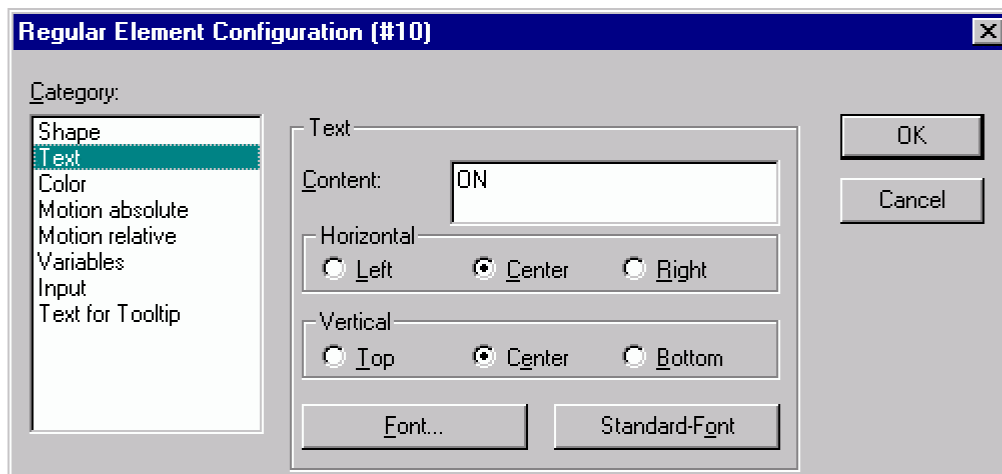


Figura 3-23. Campo de entrada para Conteúdo na categoria Texto (chave ON)

Para configurar a variável ON para TRUE com um clique do mouse na chave, ativa-se a opção 'TAP variável' na categoria 'Entrada' e entra-se ali com o nome da variável ".ON". TAP variável significa que, quando um clique do mouse é feito no elemento de visualização da variável, .ON é definida para o valor TRUE, mas volta para o valor FALSE quando o botão do mouse é solto novamente (foi criado um simples dispositivo pulsador para o programa de sinal de trânsito).

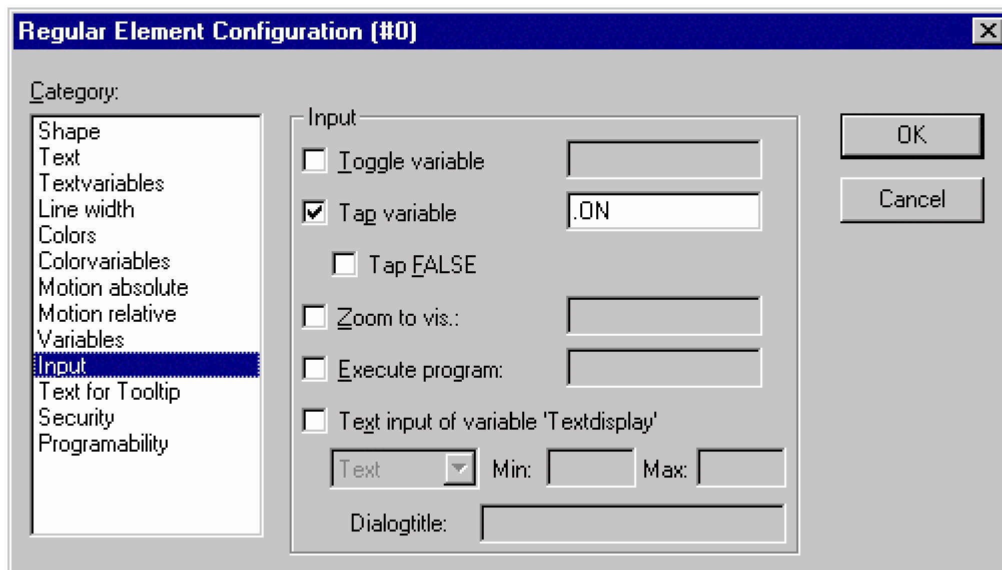


Figura 3-24. Opção 'TAP variável' na categoria 'Entrada' para definir “ON”

Fonte na visualização

Para completar a tela deve-se inserir mais dois retângulos, os quais devem ser posicionados embaixo dos sinais de trânsito.

Nas caixas de diálogo das telas configura-se branco na categoria **Cor** para **Moldura** e escreve-se na categoria **Texto**, no campo **Conteúdos**, "Light1" e "Light2". Agora a tela se parece com o que segue:

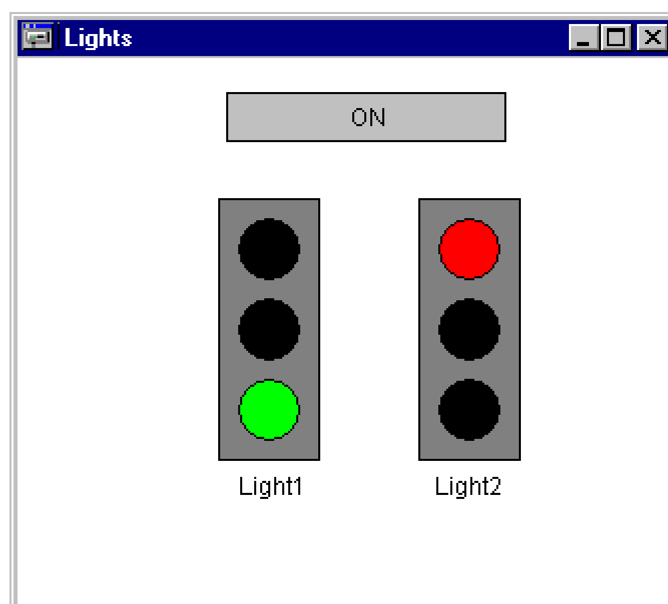


Figura 3-25. Tela resultante

4. Os componentes individuais

A janela principal

Componentes da janela principal

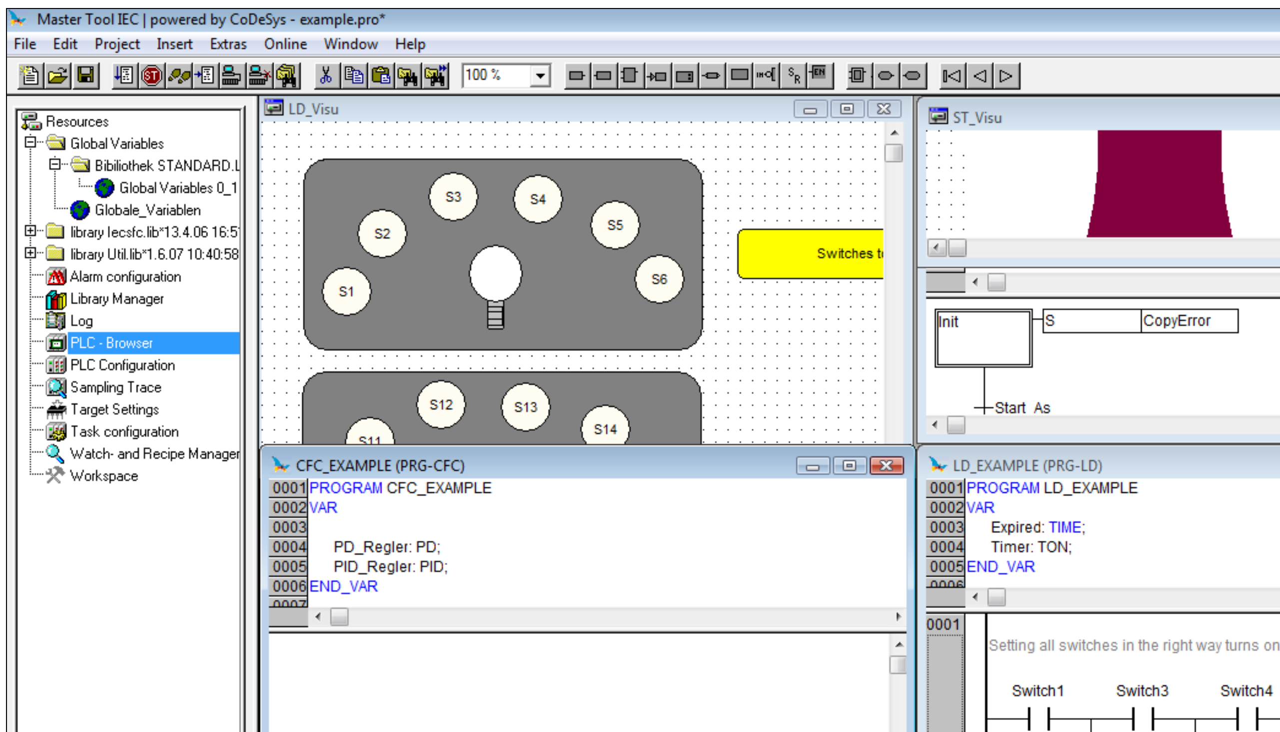


Figura 4-1. Componentes da janela principal

Os seguintes elementos são encontrados na janela principal do **MasterTool IEC** (de cima para baixo):

- A Barra do menu;
- A Barra de ferramentas (opcional); com botões para seleção mais rápida dos comandos do menu;
- O Organizador de objetos com cartões de registro para POUs, Tipo de dados, Visualizações e Recursos;
- Um divisor de tela vertical entre o Organizador de objetos e a Área de trabalho do MasterTool IEC;
- A Área de trabalho na qual as janelas do editor estão localizadas;
- A Janela de mensagens (opcional);
- A Barra de status (opcional); com informações do status do projeto.

Veja também: Menu de contexto

Barra de menu

A barra de menu está localizada na borda superior da janela principal. Ela contém todos os comandos do menu.



Figura 4-2. Barra de menu

Barra de ferramentas

Clicando com o mouse em um símbolo pode-se selecionar um comando do menu mais rapidamente. Na barra de ferramentas da janela ativa encontram-se os símbolos disponíveis.

O comando só é carregado quando o botão do mouse está pressionado no símbolo e, então, é solto.

Ao segurar o ponteiro do mouse por um tempo curto em um símbolo na barra de ferramentas, então o nome do símbolo é mostrado em uma Tooltip.

Para ver uma descrição de cada símbolo na barra de ferramentas, selecione na Ajuda o editor sobre o qual deseja-se informações e clique no símbolo da barra de ferramentas naquele em que especificamente deseja-se informações.

A exibição da barra de ferramentas é opcional (veja' Projeto' 'Opções' categoria Ambiente de trabalho).

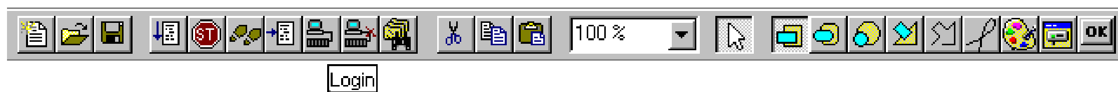


Figura 4-3. Barra de ferramentas

Organizador de objetos

O Organizador de objetos está localizado no lado esquerdo do **MasterTool IEC**. No topo há quatro cartões de registro com símbolos de quatro tipos de objetos:

POUs,

Tipo de dados,

Visualizações e

Recursos

Para alternar entre os tipos de objetos clique com o mouse no cartão de registro correspondente ou use as teclas de seta esquerda ou direita.

O capítulo Gerenciando objetos em um projeto detalha como trabalhar com os objetos no Organizador de objetos.

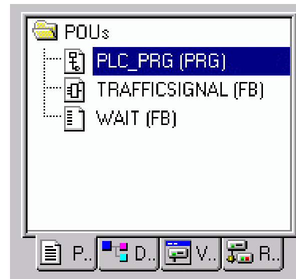


Figura 4-4. Detalhe do organizador de objetos

Divisor de telas

O divisor de tela é a borda entre duas janelas sobrepostas. No **MasterTool IEC** há divisores de telas entre o Organizador de objetos e a Área de trabalho da janela principal, entre a interface (parte das declarações) e a implementação (parte das instruções) das POUs e entre a Área de trabalho e a Janela de mensagens.

Pode-se mover o divisor de tela com o ponteiro do mouse. Isso é feito movendo o mouse com o botão esquerdo pressionado.

Assegure-se de que o divisor de tela sempre permaneça em sua posição absoluta, mesmo quando o tamanho da janela for alterado. Se parecer que o divisor de tela não está mais presente, então, simplesmente, alargue a janela.

Área de trabalho

A Área de trabalho está localizada no lado direito da janela no **MasterTool IEC**. Todos os editores para objetos e o gerenciador de bibliotecas estão abertos nesta área. O nome do objeto atual aparece na barra de título; no caso de POUs, uma abreviação para o tipo da POU e a linguagem de programação atualmente em uso aparecem entre parênteses ao final.

Uma descrição dos editores encontra-se no capítulo Os Editores.

Abaixo do item do menu **'Janela'** estão todos os comandos para gerenciamento de janelas.

Janela de mensagens

A janela de mensagens é separada por um divisor de tela abaixo da área de trabalho na janela principal.

Ela contém todas as mensagens a partir das compilações prévias, verificações ou comparações. Busca de resultados e lista de referências cruzadas também podem ser apresentadas aqui.

Ao dar um clique duplo com o mouse na janela de mensagens em uma mensagem ou pressionar <Enter>, o editor abre com o objeto. A linha relevante do objeto é selecionada. Com os comandos 'Editar' 'Próximo Erro' e 'Editar' 'Erro Anterior' pode-se rapidamente saltar entre mensagens de erro.

A exibição da janela de mensagens é opcional (veja 'Janela' 'Mensagens').

Barra de status

A barra de status no topo da moldura da janela principal no **MasterTool IEC** fornece informações sobre o projeto atual e sobre os comandos do menu.

Se um item é relevante, então o conceito aparece no lado direito da barra de status em escrita preta ou cinza.

Quando se está trabalhando no modo online, o conceito **Online** aparece em escrita preta. Trabalhando no modo offline, ele aparece em escrita cinza.

No modo **online** pode-se ver a partir da barra de status se está em simulação (**SIM**), se o programa está sendo processado (**Em execução**), se um breakpoint está configurado (**BP**), ou se as variáveis estão sendo forçadas (**Forçado**).

Com o editor de texto, o número da linha e da coluna da posição atual do cursor é indicado. (exemplo, **linha: 5**, **coluna 11**). No modo Online **'OV'** é indicado em preto na barra de status. Ao pressionar a tecla <Ins> os modos Sobrescrever e Inserir alternam-se.

Se o ponteiro do mouse está em uma tela, é fornecida a **posição** atual de **X** e **Y** em pixels, em relação ao canto esquerdo superior da tela. Se o ponteiro está em um **elemento** ou se um elemento está sendo processado, então seu número é indicado. Se há um elemento para inserir, então ele aparece também (por exemplo, **retângulo**)

Se foi escolhido um comando no menu, mas ainda não foi confirmado, então, uma breve descrição aparece na barra de status.

A exibição da barra de status é opcional (veja 'Projeto' 'Opções' categoria Ambiente de trabalho).

Menu de contexto

Atalho: <Shift> + <F10>

Em vez de usar a barra de menu para executar um comando, pode-se usar o botão direito do mouse. O menu que aparece contém os comandos mais freqüentemente usados para um objeto selecionado ou para um editor ativo. A escolha dos comandos disponíveis adapta-se automaticamente à janela ativa.

Opções do projeto

Menu de opções do projeto

Com esse comando a caixa de diálogo para configurar as opções é aberta. As opções são divididas em diferentes categorias. Escolha a categoria desejada no lado esquerdo da caixa de diálogo com um clique no mouse ou usando as teclas de setas e altere as opções no lado direito.

Uma imagem das opções que estão configuradas para o projeto atual será encontrada na guia de Recursos no componente 'Área de trabalho'.

As configurações, entre outras coisas, servem para configurar a tela principal. Elas são, exceto se determinado o contrário, salvas no arquivo .ini e restauradas na próxima inicialização do MasterTool IEC.

	Armazenado no MasterTool IEC	Armazenado no Projeto
Carregar & Salvar	x	
Informações do usuário	x	
Editor	x	
Ambiente de trabalho	x	
Cor	x	
Diretórios	Cat. Comum	Cat. Projeto
LOG	x	
Compilar		
Senhas		
Download de fonte	x	
Configuração de simbólicos	x	
Controle de fonte de projeto		x
Macros		x

Tabela 4-1. Menu de opções do projeto

Opções para carga e salvamento

Ao escolher essa categoria na caixa de diálogo Opções, então obtém-se a seguinte caixa de diálogo:

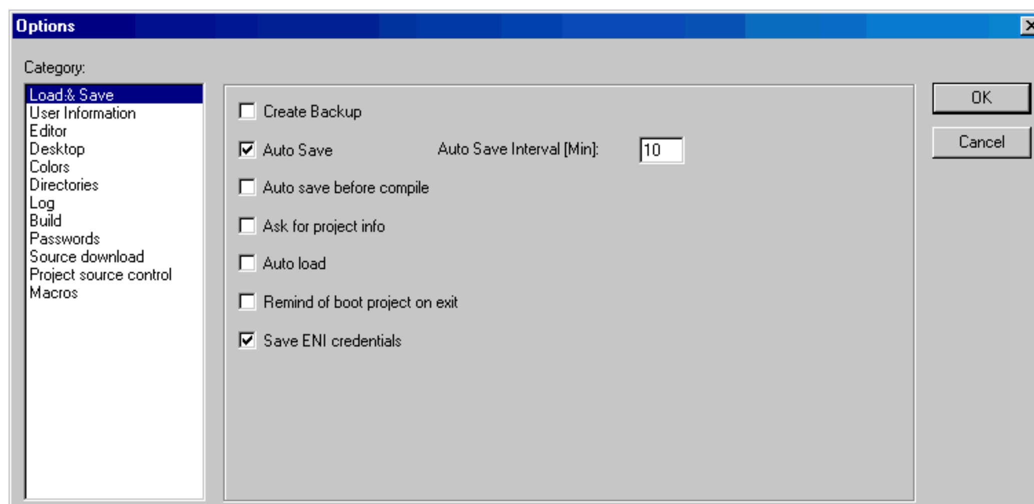


Figura 4-5. Caixa de diálogo Opções, categoria Carregar & Salvar

Quando ativar uma opção, um símbolo de seleção (✓) aparece antes da mesma.

Criar backup: o MasterTool IEC cria um arquivo de backup a cada salvamento com a extensão".bak". Ao contrário do arquivo *.asd (veja abaixo, 'Auto-Salvar') este arquivo *.bak é mantido após o fechamento do projeto. Então pode-se restaurar a versão anterior ao último projeto salvo.

Auto-salvar: enquanto se está trabalhando, o projeto é salvo de acordo com um intervalo de tempo definido (**Intervalo de auto salvamento**) em um arquivo temporário com a extensão ".asd" no diretório do projeto. Este arquivo é apagado na saída normal do programa. Se, por alguma razão, o MasterTool IEC não é fechado "corretamente" (por exemplo, devido a uma queda de energia), então o arquivo não é apagado. Quando abrir o arquivo novamente, a seguinte mensagem aparece:

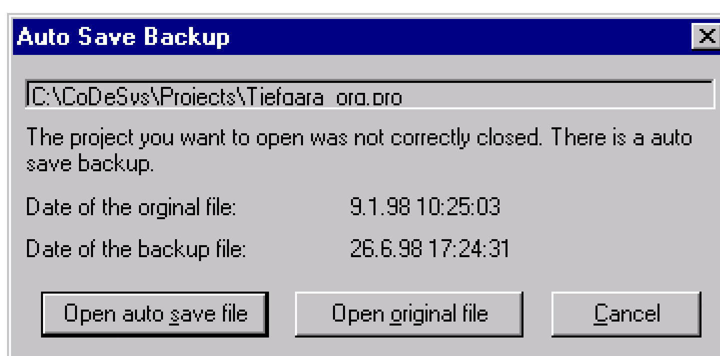


Figura 4-6. Auto-salvar

Pode-se agora decidir se quer abrir o arquivo original ou aquele que foi salvo automaticamente.

Se uma biblioteca*.lib está aberta como projeto, um arquivo de auto-salvamento correspondente "*.asl" será criado.

Auto-salvar antes de compilar: o projeto será salvo antes de cada compilação. Ao fazer isso, um arquivo com a extensão ".asd" e ".asl" será criado, o qual se comporta como descrito acima para a opção 'Auto-salvar'.

Pergunte para informações do projeto: ao salvar um novo projeto ou salvando-o com um novo nome, as informações do projeto são automaticamente chamadas. Pode-se visualizar e também processar as informações do projeto com o comando 'Projeto' 'Informações do Projeto'.

Auto-carregar: na próxima inicialização do MasterTool IEC, o último projeto aberto é automaticamente carregado. O carregamento de um projeto no início do MasterTool IEC pode ocorrer entrando com o projeto na linha de comando.

Lembrar do projeto de inicialização na saída: se o projeto foi modificado e submetido a download sem criar um novo projeto de inicialização desde o último download de um projeto de inicialização, então um diálogo irá advertir o usuário antes de sair do projeto: ”Nenhum projeto de inicialização criado desde o último download. Sair mesmo assim?”

Opções para informações de usuário

Ao escolher essa categoria na caixa de diálogo Opções, então obtém-se a seguinte caixa de diálogo.

Caixa de diálogo Opções, categoria Informações do Usuário:

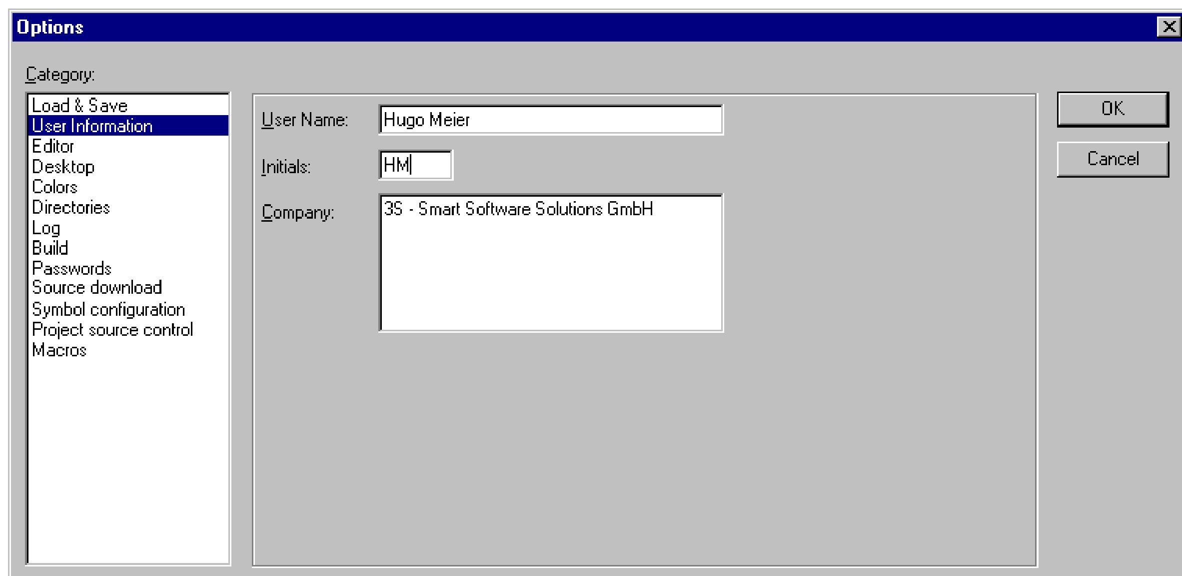


Figura 4-7. Caixa de diálogo Opções, categoria Informações do usuário

O **Nome** do usuário, suas **Iniciais** e a **Empresa** para qual trabalha constituem as Informações do usuário aqui configuradas. Cada uma das entradas pode ser modificada. As configurações serão aplicadas a quaisquer outros projetos que forem criados com o MasterTool IEC no computador local.

Opções para editor

Ao escolher essa categoria na caixa de diálogo Opções, então obtém-se a caixa de diálogo mostrada abaixo.

Pode-se fazer as seguintes configurações para os Editores:

Auto-declarar: se esta opção está ativada, então, após a entrada de uma variável ainda não declarada, uma caixa de diálogo aparecerá em todos os editores, com os quais a variável pode ser declarada.

Auto-formatar: se esta opção está ativada, então o MasterTool IEC executa uma formatação automática no editor IL e no editor de declaração. Ao finalizar uma linha, a seguinte formatação é feita:

1. Operadores escritos em letras minúsculas são mostrados em maiúsculas;

2. Guias são inseridas para que as colunas sejam divididas uniformemente.

Lista de componentes: se esta opção está ativada, então a funcionalidade "**Intellisense"** estará disponível para funcionar como um assistente de entrada. Isso significa que, ao inserir um ponto em uma posição onde um identificador deve ser inserido, então uma lista de seleção será aberta, oferecendo variáveis globais, as quais são encontradas no projeto. Ao inserir o nome de uma instância de bloco funcional, então aparecerá uma lista de seleção de todas as entradas e saídas do bloco funcional instanciado. A função Intellisense está disponível nos editores, nos gerenciadores específicos, nas telas e no rastreamento de amostragem.

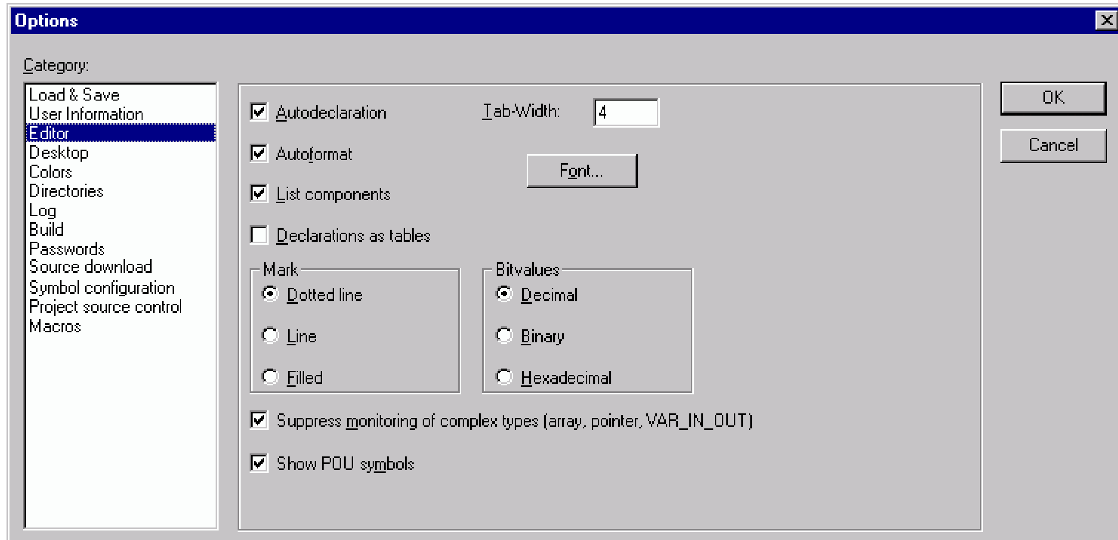


Figura 4-8. Caixa de diálogo Opções, categoria Editor

Declarações como tabelas: se esta opção está ativada, então pode-se editar variáveis em tabelas em vez de usar o editor de declaração usual. Essa tabela é classificada como uma caixa de cartões, onde se encontram guias para variáveis de entrada, variáveis de saída local e variáveis in_out. Para cada variável há campos de edição para inserir **Nome**, **Endereço**, **Tipo**, **Inicial** e **Comentário**.

Largura da guia: no campo **Largura da guia** na categoria Editor da caixa de diálogo Opções pode-se determinar a largura de uma guia como mostrado nos editores. A configuração padrão é quatro caracteres enquanto a largura do caractere depende da fonte escolhida.

Fonte: clicando no botão Fonte na categoria Editor da caixa de diálogo Opções pode-se escolher a fonte em todos os editores MasterTool IEC. O tamanho da fonte é a unidade básica para todas as operações de desenho. A escolha de uma fonte maior aumenta a impressão.

Após entrar com o comando, a caixa de diálogo da fonte abre para escolher a fonte, seu estilo e tamanho.

Caixa de diálogo para configurar a fonte:

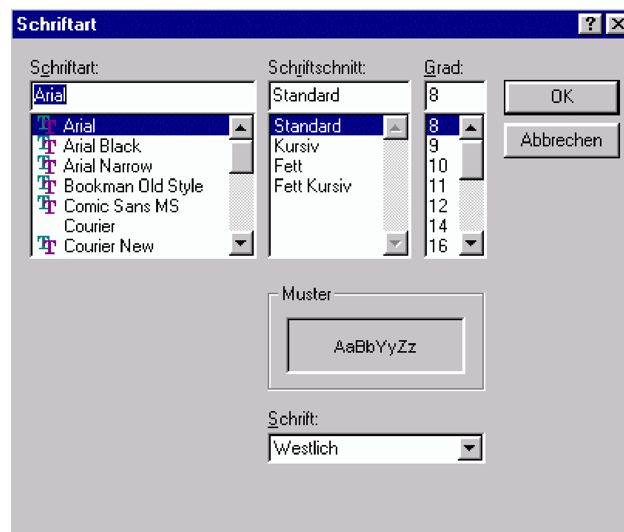


Figura 4-9. Caixa de diálogo para configurar a fonte

Marca: ao escolher Marca no na categoria Editor da caixa de diálogo Opções, pode-se optar se a seleção atual nos seus editores gráficos deve ser representada por um retângulo pontilhado (Pontilhado), um retângulo com linhas contínuas (Linha) ou por um retângulo preenchido (Preenchido). No último caso, a seleção é mostrada invertida.

Valores de bit: ao escolher Valores de bit no na categoria Editor da caixa de diálogo Opções, pode-se escolher se os dados binários (tipo BYTE, WORD, DWORD) durante a monitoração devem ser mostrados como Decimais, Hexadecimais ou Binários.

Suprimir monitoração de tipos complexos (Matriz, Ponteiro, VAR_IN_OUT): se esta opção está ativada, dados complexos como matrizes, ponteiros e VAR_IN_OUTs não serão exibidos na janela de monitoração no modo online.

Mostrar símbolos das POUs: se esta opção está ativada nas caixas de módulos que são inseridas em um editor gráfico, adicionalmente símbolos serão exibidos, se eles estiverem disponíveis na pasta da biblioteca como bitmaps. O nome do arquivo bitmap deve ser composto do nome do módulo e a extensão ".bmp". Exemplo: para o módulo TON há um arquivo símbolo TON.bmp disponível.

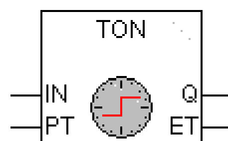


Figura 4-10. Símbolo TON

Opções para o Ambiente de trabalho

Ao escolher esta categoria em Opções na caixa de diálogo, então obtém-se a seguinte caixa de diálogo.

Quando uma opção está ativada, uma verificação aparece em frente à mesma.

Barra de ferramentas: a barra de ferramentas com os botões para seleção mais rápida dos comandos do menu está disponível abaixo da barra do menu.

Barra de status: a barra de status na borda inferior da janela principal do MasterTool IEC torna-se disponível.

Online em modo de segurança: no modo online com os comandos 'Run', 'Stop', 'Reset', 'Alternar breakpoint', 'Ciclado', 'Escrever valores', 'Forçar valores' e 'Liberar forçado(s)', uma caixa de diálogo aparece com a confirmação de que a solicitação deve realmente ser executada. Se um diálogo estendido deve estar disponível ao carregar o projeto atual a partir do sistema de programação para o

CP, se isso for suportado pelo sistema do dispositivo. Se já houver um projeto no CP, esse diálogo exibirá as informações desse projeto, assim como as informações de um projeto de inicialização quando já houver um no CP. Esta opção é salva com o projeto.

Caixa de diálogo Opções, categoria Ambiente de trabalho:

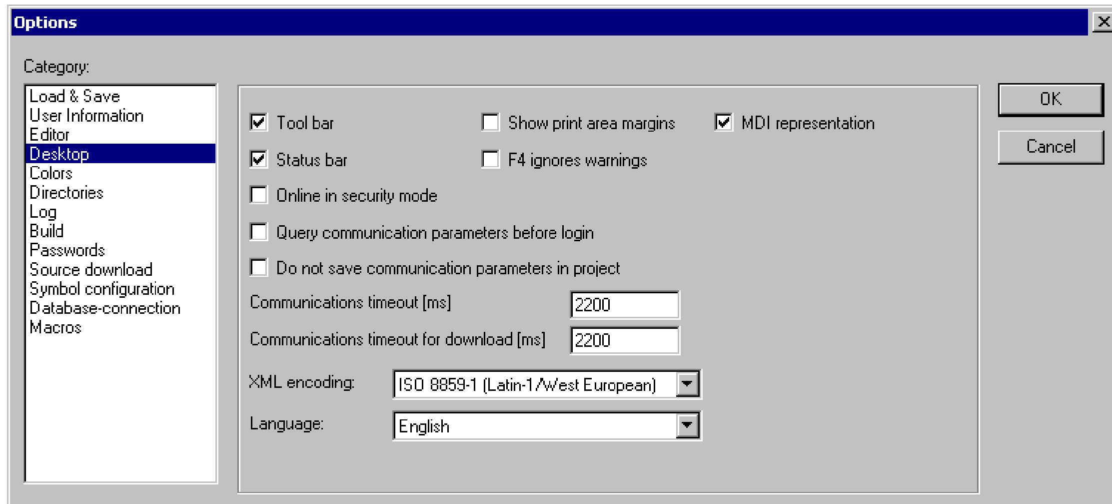


Figura 4-11. Caixa de diálogo Opções, categoria Ambiente de trabalho

Parâmetros de comunicação query antes do login: assim que o comando 'Comunicação' 'Login' é executado, primeiro o diálogo dos parâmetros de comunicação abrirá. Para entrar no modo online deve-se primeiro fechar este diálogo com OK.

Não salvar os parâmetros de comunicação no projeto: as configurações do diálogo dos parâmetros de comunicação ('Comunicação' 'Parâmetros de comunicação') não serão salvos com o projeto.

Mostrar margens da área de impressão: em cada janela do editor, os limites do intervalo atual de impressão configurado são marcados com linhas vermelhas tracejadas. O tamanho delas depende das características da impressão (tamanho do papel, orientação) e do tamanho do campo de "Conteúdo" do layout atual de impressão configurado.

F4 ignora advertências: após a compilação, quando F4 é pressionado em uma janela de mensagens, o foco pula somente para linhas com mensagens de erro; mensagens de advertência são ignoradas.

Representação MDI: por padrão, esta opção (**M**ultiple-**D**ocument-**I**nterface) é ativada e, desta maneira, muitas janelas podem ser abertas ao mesmo tempo. Se esta opção está desativada (modo SDI), somente uma janela pode ser aberta e será exibida no modo tela cheia. Exceção: a ação de um programa e do programa em si pode ser exibida lado a lado mesmo neste modo.

Timeout de comunicação [ms]: para serviços de comunicação padrão é o tempo específico, em milissegundos, após o qual a comunicação com o sistema do dispositivo será concluída se não for detectada mais nenhuma atividade. Valores possíveis: 1-10000000 ms.

Timeout de comunicação para download [ms]:para serviços de comunicação de longa duração (download de programa, upload/download de arquivo, criação e verificação de projeto de inicialização) é o tempo específico, em milissegundos, após o qual a comunicação com o sistema do dispositivo será concluída se não for detectada mais nenhuma atividade (Tempo de espera de download). Valores possíveis: 1-10000000 ms.

Idioma: defina aqui o idioma do menu e quais textos de diálogo devem ser exibidos.

Nota: considerar que a escolha do idioma não é possível no Windows 98T!

Nota: as configurações serão armazenadas no MasterTool IEC.

Opções para cores

Ao escolher esta categoria na caixa de diálogo Opções, então obtém-se o que segue.

Caixa de diálogo Opções, categoria Cores:

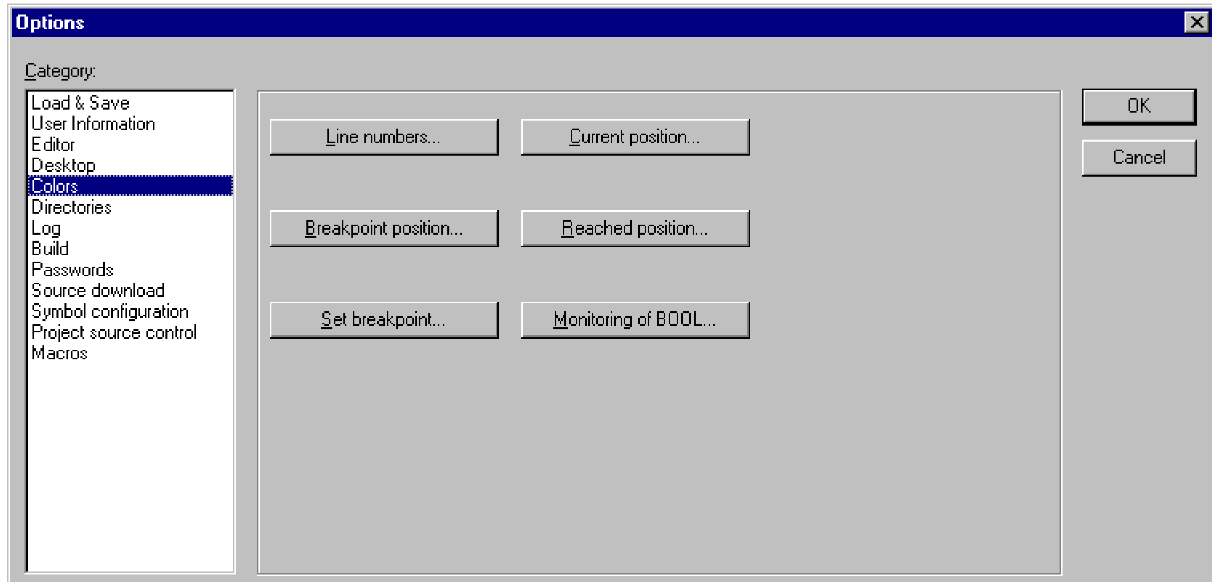


Figura 4-12. Caixa de diálogo Opções, categoria Cores

Pode-se editar a configuração de cor padrão do MasterTool IEC. Também pode-se escolher se deseja-se alterar as configurações de cor para **Números de linhas** (padrão pré-definido: verde claro), para **posições de Breakpoint** (cinza escuro), para **Definir breakpoint** (azul claro), para a **Posição atual** (vermelho), para as **Posições alcançadas** (verde) ou para a **Monitoração de variáveis booleanas** (azul).

Ao escolher um dos botões indicados, a caixa de diálogo para a entrada de cores abre.

Caixa de diálogo para configuração de cores:



Figura 4-13. Caixa de diálogo para configuração de cores

Opções para diretórios

Ao escolher esta categoria na caixa de diálogo Opções, então obtém-se a seguinte caixa de diálogo:

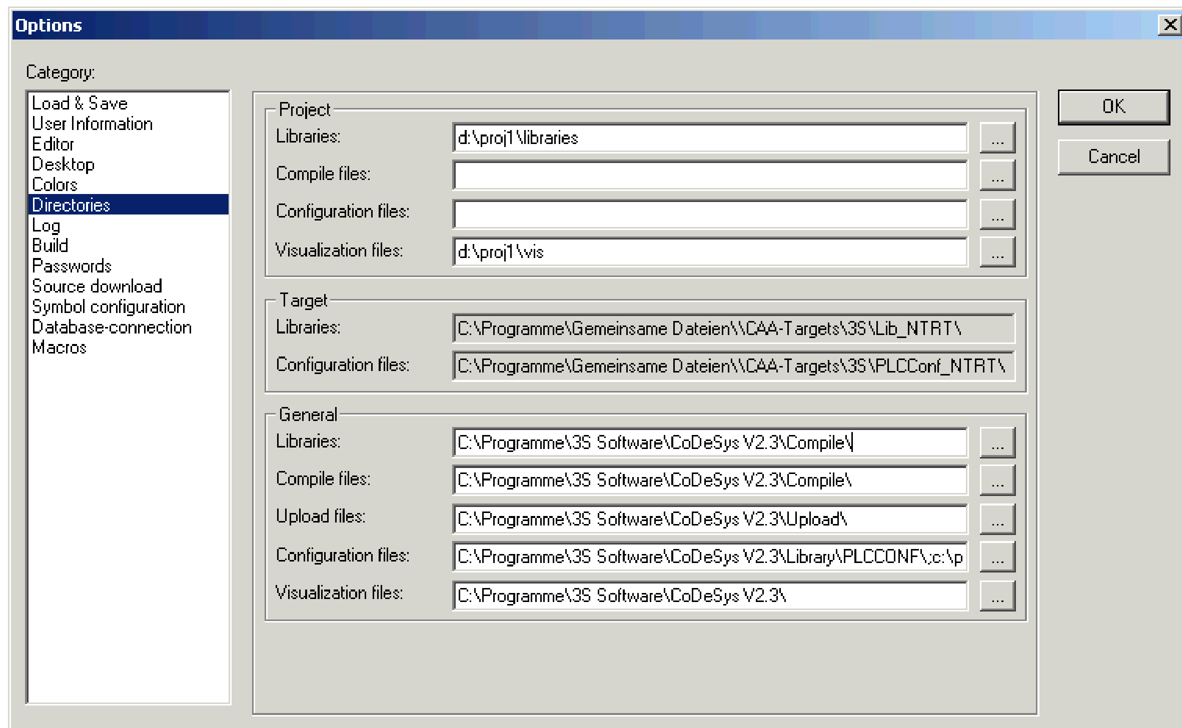


Figura 4-14. Caixa de diálogo Opções, categoria Diretórios

Diretórios podem ser inseridos no **Projeto** e nas **Áreas Comuns** para que o MasterTool IEC possa localizar **Bibliotecas**, arquivos de **Configuração** e **Visualização** do controlador (bitmaps, arquivos XML para textos dinâmicos, etc.), assim como para armazenar arquivos de **compilação** e **upload** de fonte (arquivos de compilação, por exemplo, são arquivos map e list, mas não são, por exemplo, arquivos de símbolos! Os últimos serão salvos no diretório do projeto). Ao ativar o botão (...), o diálogo de seleção do diretório abre. Para bibliotecas e arquivos de configuração, vários caminhos podem ser usados para cada um, separados por ponto e vírgula “;”.

Considerar:

- Caminhos de bibliotecas podem ser inseridos baseados no caminho do arquivo do projeto, colocando como prefixo um ponto ".". Se, por exemplo, ".\libs" for inserido, as bibliotecas serão localizadas em 'C:\programs\projects\libs', se o projeto atual está em 'C:\programs\projects' para informações sobre caminhos para bibliotecas, veja também: 'Inserir' 'Biblioteca adicional'.

- Não use espaços vazios e caracteres especiais, exceto "_" nos caminhos dos diretórios.

Informação: na área **Projeto**, as configurações serão armazenadas no projeto.

Informações na área **Comum** estão gravadas no arquivo-ini do sistema de programação.

A área do **dispositivo** exibe os diretórios para bibliotecas e arquivos de configuração configurados no sistema do dispositivo, por exemplo, através de entradas no arquivo do dispositvo. Esses campos não podem ser editados, mas uma entrada pode ser selecionada e copiada (botão direito do mouse no menu de contexto).

O MasterTool IEC geralmente busca primeiro nos diretórios inseridos em 'Projeto', depois naqueles em 'Dispositivo' (definidos no arquivo do dispositivo) e, finalmente, naqueles listados sob o nome 'Comum'. Se dois arquivos com nomes idênticos forem encontrados, será usado aquele no diretório que for localizado antes.

Opções para compilação

Ao escolher esta categoria na caixa de diálogo Opções, então obtém-se a caixa de diálogo mostrada abaixo:

Depuração: depende das descrições do dispositivo se essa opção pode ser ativada/desativada. Se estiver ativada, um código de depuração adicional será criado, ou seja, o código pode se tornar consideravelmente maior. O código de depuração é necessário para fazer uso das funções de depuração oferecidas pelo MasterTool IEC (por exemplo, breakpoints). Ao desativar essa opção, o processamento do projeto se torna mais rápido e diminui o tamanho do código. A opção é armazenada com o projeto.

Substituir constantes: o valor de cada constante do tipo escalar (excluindo, portanto, STRING, matriz e estrutura) é carregado diretamente e, no modo online, as constantes são exibidas na cor verde. Forçar, escrever e monitorar uma constante, então, não é mais possível. Se a opção estiver desativada, o valor será carregado em um local de armazenamento por meio de variáveis de acesso (isso na verdade permite escrever o valor da variável, mas implica em um tempo de processamento maior).

Comentários aninhados: comentários podem ser colocados dentro de outros comentários. Exemplo:

```
(*
A:=INST.OUT; (* A SER VERIFICADO *)
B:=B+1;
*)
```

Aqui o comentário que começa com a primeira abertura de parênteses não é fechado logo após "VERIFICADO", mas apenas no último fechamento de parênteses.

Atenção: essa opção deve ser usada com cuidado. Se a configuração do projeto não coincidir com a configuração escolhida em uma biblioteca que também foi criada no MasterTool IEC e agora é usada no projeto, erros de compilação ocorrerão, que são difíceis de interpretar pelo usuário e geralmente não podem ser identificados!

Caixa de diálogo Opções, categoria Compilar:

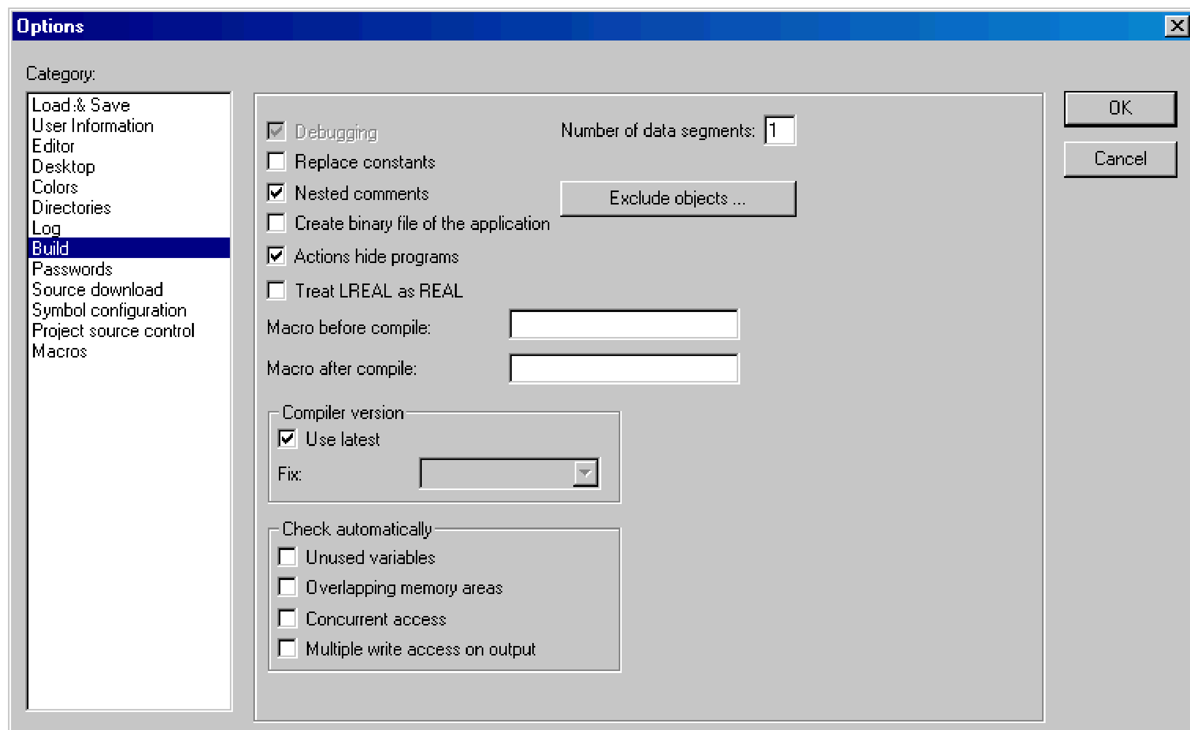


Figura 4-15. Caixa de diálogo Opções, categoria Compilar

Criar arquivo binário da aplicação: uma imagem binária do código gerado (projeto de inicialização) é criada no diretório de projeto durante a compilação. Nome do arquivo:

<nome_do_projeto>.bin. Por comparação, o comando 'Comunicação' 'Criar projeto de inicialização' configura o projeto de inicialização no controlador.

Ações ocultar programas: quando um novo projeto é criado, essa opção é ativada por padrão. Isso significa que, se uma ação local tiver o mesmo nome, como uma variável global ou um programa, vale a seguinte hierarquia: variável local antes da ação local, antes da variável global, antes do programa.

Considerar: se é aberto um projeto existente criado com uma versão anterior do MasterTool IEC, a opção será desativada por padrão. Assim, a hierarquia anteriormente válida (variável local, antes da variável global, antes do programa, antes da ação local) pode ser mantida.

Tratar LREAL como REAL: se esta opção é ativada (a disponibilidade depende do controlador; padrão: não ativado), a compilação tratará valores LREAL como valores REAL. Isso pode ser usado para a criação de plataformas de projetos independentes.

Número de segmentos de dados: aqui se define quantos segmentos de memória devem ser alocados no CP para os dados do projeto. Esse espaço é necessário para tornar possíveis alterações online, mesmo se tiverem sido adicionadas novas variáveis. Se durante a compilação aparecer a mensagem "Fora da memória de dados globais...", digite um número mais alto. Nesse aspecto, variáveis locais de programa serão tratadas como variáveis globais.

Excluir objetos: este botão abre o diálogo **excluir objetos da compilação**. Na árvore de componentes de projeto selecione as POUs que não devem ser consideradas durante a compilação e ative a opção **Excluir**. Desta forma, as POUs excluídas serão exibidas na cor verde na árvore de seleção. Pressione o botão **Excluir não utilizados**, para exibir apenas as POUs que estão sendo usadas no programa no momento. Considere que um único objeto selecionado na biblioteca de objetos pode também ser excluído da compilação, usando o comando **'Excluir da compilação'** no menu de contexto.

Versão do compilador: aqui se define a versão do compilador a ser usado. Versões MasterTool IEC após V2.3.3 incluirão, além da versão do compilador atual, também as versões anteriores do compilador (anterior à V2.3.3). Para que o projeto seja compilado com a versão mais recente do compilador, em qualquer caso, ative a opção **Usar mais recentes**. Nesse caso, entretanto, será verificado se o sistema de programação aberto no momento é também o daquela versão. Se não for, a versão do compilador correspondente a versão atual do sistema de programação em uso será considerada! Se o projeto deve ser compilado com uma versão específica, defina isto por meio da lista de seleção em Corrigir.

Para controlar o processo de compilação, pode-se configurar duas macros:

A macro no campo **Macro antes de compilar** é executada antes do processo de compilação; a macro no campo **Macro depois de compilar**, é executada após a operação de compilação. Os seguintes comandos de macro não podem, no entanto, ser usados aqui: arquivo novo, abrir arquivo, fechar arquivo, salvar arquivo como, arquivo sair, online, compilar projeto, verificar projeto, limpar projeto, compilar tudo, depurar e lista de monitoração.

Verificar automaticamente: para obter a correção semântica marcada em cada compilação do projeto, as opções a seguir podem ser ativadas:

- Variáveis não utilizadas
- Sobreposição de áreas de memória
- Acesso concorrente
- Acesso de escrita múltipla na saída

Os resultados serão exibidos na janela de mensagens. Essas verificações também podem ser iniciadas pelos respectivos comandos do sub-menu 'Verificar' no menu 'Projeto'.

Se suportado pelo sistema do dispositivo, resultados negativos de seleção produzirão erros de compilação.

Considerar: todas as entradas no diálogo de opções de compilação são armazenadas com o projeto.

Senhas

Para proteger seus arquivos contra acessos não autorizados, o MasterTool IEC oferece a opção de usar uma senha para evitar que seus arquivos sejam abertos ou alterados.

Ao escolher esta categoria na caixa de diálogo Opções, então obtém-se a caixa de diálogo correspondente.

Insira a senha desejada no campo **Senha**. Para cada caractere digitado um asterisco aparece no campo. Deve-se repetir a mesma palavra no campo **Confirmar senha**. Feche a caixa de diálogo com **OK**. Ao receber a mensagem:

“A senha não confere com a confirmação", então, ocorreu um erro de digitação durante uma das duas entradas. Nesse caso, repita-as até que a caixa de diálogo seja fechada sem nenhuma mensagem.

Após salvar o arquivo e reabri-lo, obtém-se uma caixa de diálogo na qual será solicitada a senha. O projeto somente poderá ser aberto se for inserida a senha correta. Caso contrário, o MasterTool IEC informa:

"A senha não está correta."

Caixa de diálogo Opções da categoria Senhas:

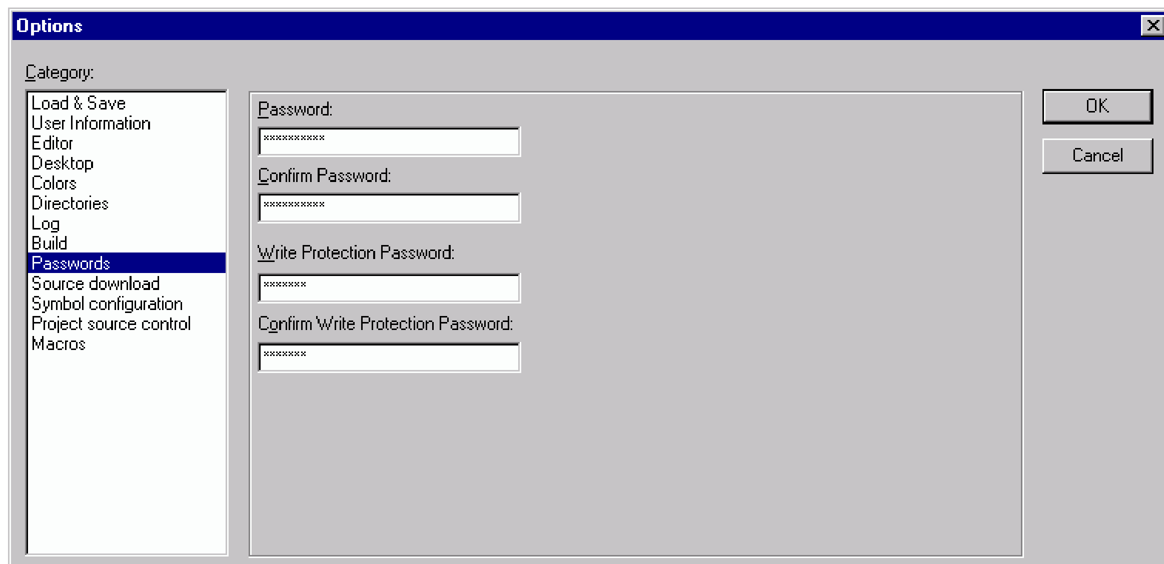


Figura 4-16. Caixa de diálogo Opções da categoria Senhas

Juntamente com a abertura do arquivo, também pode-se usar uma senha para proteger o arquivo que está sendo alterado. Para isso deve-se inserir uma senha no campo **Escrever senha de proteção** e confirme essa entrada no campo seguinte.

Um projeto protegido contra gravação pode ser aberto sem uma senha. Para isto, simplesmente pressione o botão **Cancelar**, quando o MasterTool IEC solicitar a digitação da senha de proteção contra gravação ao abrir um arquivo. Nesse caso, poder-se-á compilar o projeto, carregá-lo para o CP, simulá-lo, etc., mas não poderá alterá-lo.

Naturalmente é importante memorizar as duas senhas. No entanto, em caso de esquecimento, contate a ALTUS.

As senhas são salvas com o projeto.

Para criar direitos de acesso diferenciados pode-se definir grupos de usuários e "Senhas para grupos de usuários").

Além disso, considere as possibilidades estendidas para proteger um projeto por criptografia (consultar nesse mesmo capítulo 'Arquivo' 'Salvar como') a qual, por exemplo, pode ajudar a proteger o uso de uma biblioteca sem que tenha sido inserida uma senha.

Download de fonte

Ao escolher esta categoria na caixa de diálogo Opções, então obtém-se o que segue:

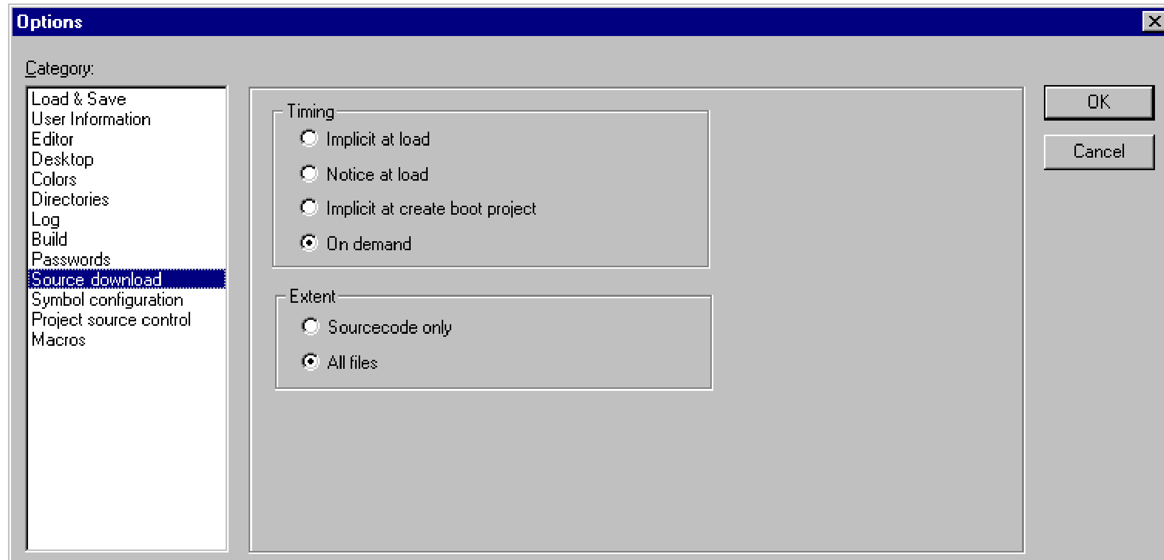


Figura 4-17. Caixa de diálogo Opções, categoria Download de fonte

Pode-se optar para qual **Timing** e qual **Extensão** o projeto é carregado para o sistema do controlador. A opção **Código fonte** envolve **exclusivamente** o arquivo MasterTool IEC (arquivo de extensão .pro). A opção **Todos os arquivos** também inclui arquivos como os arquivos associados às bibliotecas, teelas, arquivos de configuração, etc..

A opção **Implícito na carga** permite que o arquivo selecionado seja carregado automaticamente para o sistema do controlador no comando **'Comunicação'** 'Download'.

A opção **Implícito ao criar projeto de inicialização** permite que o intervalo de arquivo selecionado seja automaticamente carregado para o sistema do controlador no comando 'Comunicação' 'Criar projeto de inicialização'.

A opção de **Aviso na carga** oferece uma caixa de diálogo, quando o comando **'Comunicação'** 'Download' for especificado, com a pergunta "Você deseja gravar o código fonte no sistema do controlador?". Pressionando **Sim** o intervalo selecionado de arquivos para o sistema do controlador será automaticamente carregado, ou pode-se alternativamente finalizar com **'Não'**.

Quando a opção **Sob demanda** estiver acionada, o intervalo selecionado de arquivos será expressamente carregado no sistema do controlador via comando 'Comunicação' 'Download de código fonte'.

O projeto armazenado no sistema do controlador pode ser recuperado usando 'Abrir' 'Arquivo' com Abrir projeto a partir do CP. Os arquivos serão descompactados no processo.

Opções para Macros

Ao escolher essa categoria na caixa de diálogo Opções, é aberta uma caixa de diálogo onde pode-se definir as macros usando os comandos do mecanismo de lote do MasterTool IEC. Estas macros podem então ser chamados no menu 'Editar' 'Macros'.

Caixa de diálogo Opções da categoria Macros:

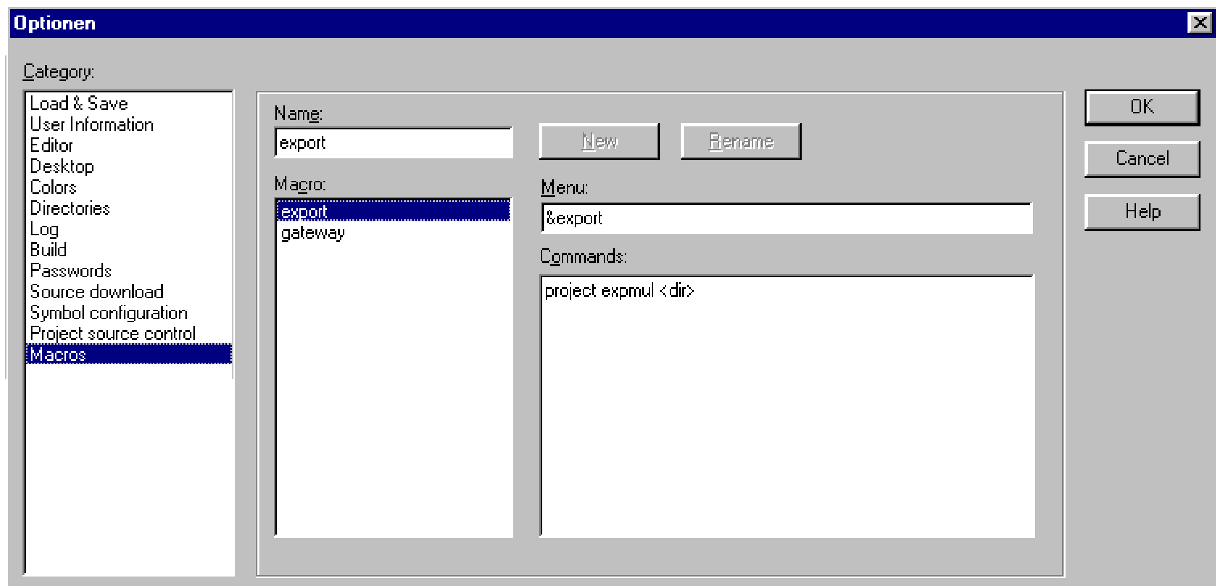


Figura 4-18. Caixa de diálogo Opções da categoria Macros

Execute as seguintes etapas para definir uma nova macro:

No campo de entrada **Nome**, insira um nome para a macro a ser criada. Após o botão **Novo** ser pressionado, este nome é transferido para o campo **Lista de macros** e lá é marcado como selecionado. A lista de macros é representada em uma estrutura de árvore. As macros localmente definidas são posicionadas uma abaixo da outra. Se bibliotecas de macro (veja abaixo) forem integradas, então, os nomes da biblioteca serão listados e, via clique do mouse nos sinais de adição e subtração na frente destas entradas, pode-se abrir ou fechar uma lista dos elementos de biblioteca.

O campo **menu** é usado para definir a entrada de menu com a qual a macro será exibida no menu 'Editar' 'Macros'. Para ser capaz de usar uma única letra como um atalho, a letra deve ser precedida pelo símbolo '&'. Exemplo: o nome "Ma&cro 1" gera a entrada de menu "Macro 1".

No campo de edição **Comandos** definem-se e/ou editam-se os comandos que constituirão a macro recém-criada ou selecionada. Todos os comandos do mecanismo de lote do **MasterTool IEC** e todas as palavras-chave que são válidas para aqueles são permitidas. Pode-se obter uma lista pressionando o botão **Ajuda**. Uma nova linha de comando é iniciada, pressionando <Ctrl>+<Enter>. O menu de contexto com as funções de editor de texto comuns é obtido pressionando o botão direito do mouse.Os componentes de comando que operam em conjunto podem ser agrupados usando aspas.

Para criar outras macros execute os passos de 1 a 3 novamente, antes de fechar a caixa de diálogo, pressionando o botão OK.

Para excluir uma macro, selecione-a na lista de macros e pressione botão <Del>.

Para renomear uma macro, selecione-a na lista de Macro, insira um novo nome no campo de edição 'Nome' e, em seguida, pressione o botão **Renomear**.

Para editar uma macro existente, selecione-a na lista de macro e edite os campos 'Menu' e / ou 'Comandos'. As modificações serão salvas quando pressionar o botão OK.

Assim que a caixa de diálogo é fechada após pressionar o botão **OK**, as descrições atuais de todas as macros serão salvas no projeto.

As entradas do menu Macro no menu **'Editar'** 'Macros' agora são exibidas na ordem em que foram definidas. As macros não são verificadas até que seja feita uma seleção de menu.

Bibliotecas de macro:

As macros podem ser salvas em bibliotecas externas de macro. Essas bibliotecas podem ser incluídas em outros projetos.

Criando uma biblioteca de macro que contém as macros do projeto atual: pressione o botão **Criar** para obter o diálogo **Mesclar projeto**, onde todas as macros disponíveis são listadas. Selecione as entradas desejadas e confirme com OK. A caixa de diálogo de seleção será fechada e a caixa de diálogo **Salvar biblioteca de macro** será aberta. Insira um nome e caminho para a nova biblioteca e pressione o botão Salvar. A biblioteca será criada denominada como < **nome da biblioteca** >.**mac** e a caixa de diálogo será fechada.

Incluindo uma biblioteca macro < nome da biblioteca >.mac no projeto atual: pressione o botão **Incluir**. A caixa de diálogo **Abrir biblioteca macro** será aberta mostrando os arquivos com extensão *. mac. Selecione a biblioteca desejada e pressione o botão **Abrir**. A caixa de diálogo será fechada e a biblioteca será adicionada à árvore da Lista de macros.

Nota: as macros de um projeto também podem ser exportadas ('Projeto' 'Exportar').

Nota: as configurações serão armazenadas no projeto.

Gerenciando projetos

Os comandos referentes ao projeto são encontrados sob os itens do menu 'Arquivo' e 'Projeto'.

Menu ‘Arquivo’ ‘Novo’

Símbolo:

Com este comando é criado um projeto vazio com o nome "Sem título". Esse nome deve ser alterado ao salvá-lo.

Menu ‘Arquivo’ ‘Novo a partir do modelo’

Use esse comando para abrir qualquer projeto desejado do MasterTool IEC como um projeto "modelo". A caixa de diálogo para abrir um arquivo de projeto estará disponível e o projeto selecionado será aberto com o nome "desconhecido".

Menu ‘Arquivo’ ‘Abrir’

Símbolo:

Com este comando abre-se um projeto já existente. Se um projeto já foi aberto e alterado, o MasterTool IEC perguntará se esse projeto deve ser salvo ou não.

A caixa de diálogo para abrir um arquivo aparece e um arquivo de projeto com a extensão "* .pro"ou um arquivo de biblioteca com a extensão"* .lib" deve ser escolhido. Esse arquivo já deve existir. Não é possível criar um projeto com o comando "**Abrir**".

Caixa de diálogo para abrir um arquivo:

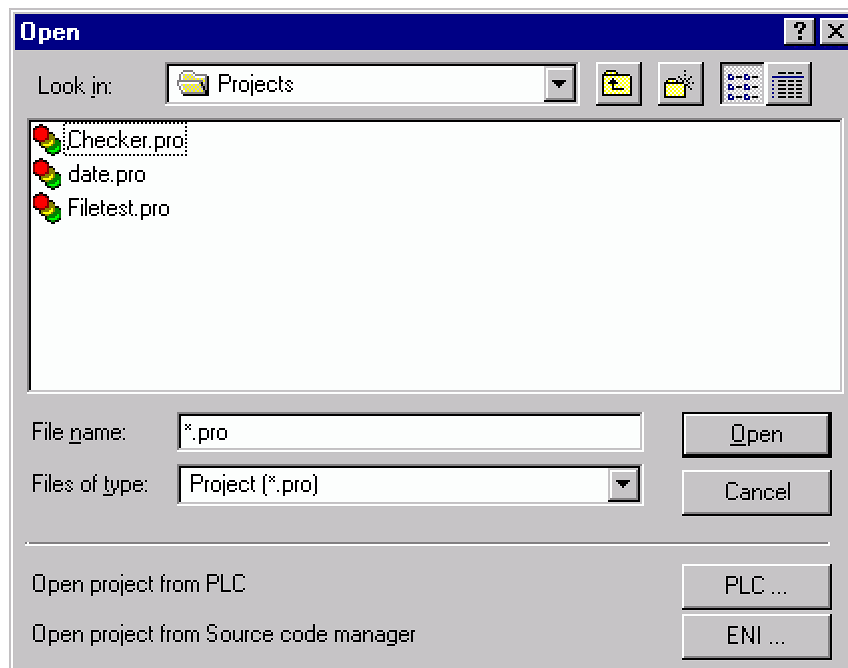


Figura 4-19. Caixa de diálogo para abrir um arquivo

Abrir um projeto a partir do CP

Para carregar um arquivo de projeto do CP, pressione CP em 'Abrir projeto a partir do CP'. Obtém-se o diálogo 'Parâmetros de comunicação' (consultar o menu 'Comunicação' 'Parâmetros de comunicação') para definir os parâmetros de transmissão quando ainda não existe conexão para o CP. Após ter sido criada uma conexão online, o sistema verifica se os mesmos arquivos de projeto nomeados já existem no diretório do seu disco rígido do computador. Quando este for o caso obtém-se a caixa de diálogo 'Carregar o projeto a partir do controlador', onde pode-se decidir se os arquivos locais devem ser substituídos por esses sendo usados pelo controlador (essa seqüência é o inverso da seqüência de 'Comunicação' 'Carregar código fonte', com o qual o arquivo de origem do projeto é armazenado no controlador). Não confunda com 'Criar projeto de inicialização'!).

Nota: por favor, observe que em qualquer caso é necessário nomear o novo projeto quando o mesmo é carregado do CP para o seu diretório local. Caso contrário, ele fica sem nome. Se suportado pelo sistema do dispositivo, um 'Título' inserido nas Informações do projeto será predefinido como novo nome do arquivo do projeto. Nesse caso, ao carregar o projeto do CP, a caixa de diálogo para salvar um arquivo será aberta, onde o novo nome do arquivo é inserido automaticamente e pode ser confirmado ou modificado.

Se o projeto ainda não foi carregado para o CP, é exibida uma mensagem de erro. Veja também 'Projeto' 'Opções' Categoria 'Download de Fonte'.

Arquivos abertos recentemente

Os arquivos mais recentemente abertos são listados no menu Arquivo abaixo do comando 'Arquivo' 'Sair'. Ao escolher um deles, então, este projeto é aberto.

Se forem definidos Senhas ou Grupos de usuários para o projeto, uma caixa de diálogo será exibida para inserir a senha.

Menu ‘Arquivo’ ‘Fechar’

Com esse comando fecha-se o projeto atualmente aberto. Se o projeto tiver sido alterado, o MasterTool IEC perguntará se essas alterações devem ser salvas ou não.

Se o projeto a ser salvo assumir o nome "Sem título”, um nome deve ser dado a ele (consultar 'Arquivo' 'Salvar como').

Menu ‘Arquivo’ ‘Salvar’

Símbolo:

Atalho: <Ctrl>+<S>

Com este comando salvam-se quaisquer alterações no projeto. Se o projeto a ser salvo é chamado "Sem título, deve-se nomeá-lo (consultar 'Arquivo' 'Salvar como').

Menu ‘Arquivo’ ‘Salvar como’

Com esse comando o projeto atual pode ser salvo em outro arquivo ou como uma biblioteca. Isso não altera o arquivo do projeto original.

Depois que o comando foi escolhido, a caixa de diálogo Salvar é exibida. Escolha um **nome de arquivo** existente ou digite um novo nome de arquivo e escolha o **tipo de arquivo** desejado.

Se o projeto está para ser salvo sob um novo nome, escolha o tipo de arquivo de projeto MasterTool IEC (* .pro).

Ao escolher o tipo de arquivo do **projeto versão 1.5 (* .pro), 2.0 (* .pro), 2.1 (* .pro) ou 2.2 (* .pro)**, então, o projeto atual é salvo como se fosse criado com a versão 1.5, 2.0, 2.1 ou 2.2. Desta forma, dados específicos da versão 2.3, podem ser perdidos! No entanto, o projeto pode ser executado com a versão 1.5, 2.0, 2.1 ou 2.2.

Caixa de diálogo Salvar como:

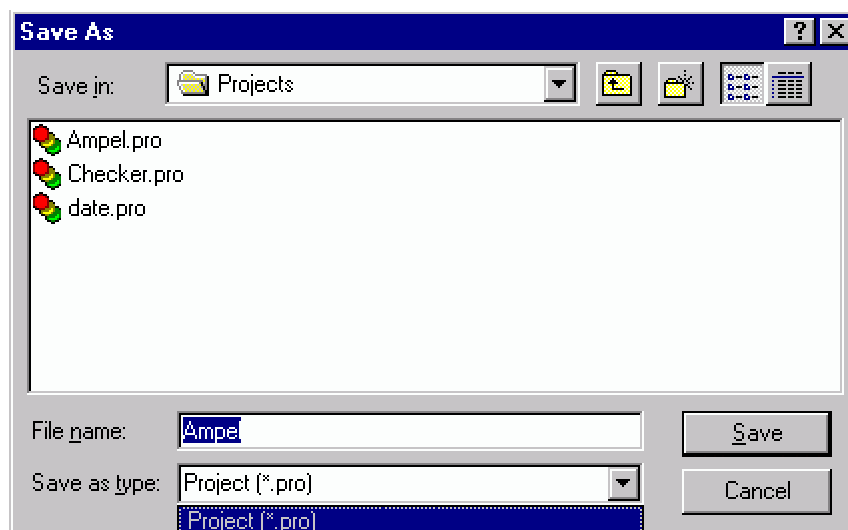


Figura 4-20. Caixa de diálogo Salvar como

É possível salvar o projeto atual como uma biblioteca para usá-lo em outros projetos. Escolha o tipo de arquivo **Biblioteca Interna** (* .lib) se as POUs foram programadas no MasterTool IEC.

Escolha o tipo de arquivo **Biblioteca Externa** (* .lib) se desejar implementar e integrar as POUs em outras linguagens (por exemplo, C). Isto significa que aquele outro arquivo também foi salvo, o qual recebe o nome do arquivo da biblioteca, mas com a extensão "*.h". Este arquivo é construído como um arquivo de cabeçalho C com as declarações de todas as POUs, tipos de dados e variáveis globais. Se bibliotecas externas são usadas no modo de simulação, a implementação escrita para os POUs no MasterTool IEC, será executada. Se trabalhar com o hardware real, a implementação escrita em C será executada.

Criptografia de um projeto

Para salvar o projeto como um projeto criptografado ou uma biblioteca, escolha uma das opções: **Projeto criptografado** MasterTool IEC (* .pro), **Biblioteca interna criptografada** (* .lib) ou **Biblioteca externa criptografada** (* .lib). Nesse caso, obtém-se a caixa de diálogo 'Criptografia',

onde pode-se definir e confirmar a chave. O projeto não pode ser aberto posteriormente e uma biblioteca não pode ser usada sem essa chave.

Caixa de diálogo para criptografar um projeto:

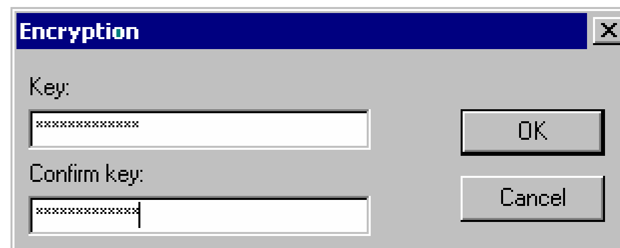


Figura 4-21. Caixa de diálogo para criptografar um projeto

A criptografia amplia a proteção de um projeto, o que até agora era possível apenas via atribuição de senhas para acesso e proteção de gravação. Essas possibilidades existirão mais adiante, mas considere que elas, por exemplo, não podem evitar que uma biblioteca seja incluída em um projeto sem a necessidade de digitar uma senha de biblioteca (chave).

Uma chave, uma vez definida, será salva com qualquer salvamento posterior do projeto. Para modificar essa chave, deve-se usar novamente a caixa de diálogo 'Salvar como'.

Se um projeto criptografado deve ser aberto e se uma biblioteca criptografada deve ser usada em um projeto, a caixa de diálogo solicitando a chave será exibida.

Caixa de diálogo para inserir a chave:

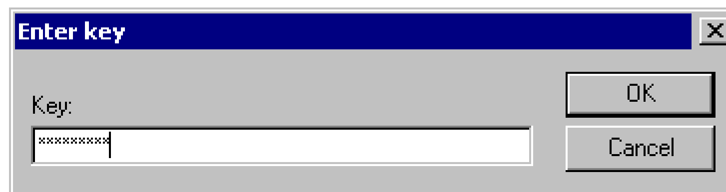


Figura 4-22. Caixa de diálogo para inserir a chave

Licenciamento de uma biblioteca:

Para salvar o projeto como uma biblioteca licenciada, pode-se adicionar as informações de licenciamento apropriadas na caixa de diálogo 'Editar informações de licença'. Abra a caixa de diálogo pressionando o botão 'Editar informações de licença'. Consultar para descrição: 'Gerenciamento de licenças no MasterTool IEC'.

Após ter feito todas as configurações, pressione ' **OK** '. O projeto atual será salvo no arquivo indicado. Se o novo nome de arquivo já existir, o MasterTool IEC perguntará se é para substituir o arquivo anterior.

No salvamento como uma biblioteca, o projeto inteiro é compilado. Se ocorrer um erro, será informado de que, para criar uma biblioteca, é necessário um projeto correto. O projeto, então, não será salvo como uma biblioteca.

Menu ‘Arquivo’ ‘Salvar/enviar arquivo’

Este comando é usado para configurar e criar um arquivo compactado de projeto. Todos os arquivos que são referenciados e usados com um projeto MasterTool IEC podem ser compactados em um arquivo ZIP. O arquivo ZIP pode ser armazenado ou enviado diretamente via e-mail. Isto é útil se deseja-se encaminhar um conjunto de todos os arquivos relevantes do projeto.

Considerar: a função de arquivamento não é prática para restaurar um ambiente de projeto. Ela é designada para uma fácil compactação de todos os arquivos pertencentes a um projeto. Ao

descompactar um arquivo, os caminhos dos arquivos determinados devem ser adaptados ao ambiente MasterTool IEC atual!

Quando o comando é executado, a caixa de diálogo Salvar arquivo abre.

Caixa de diálogo para configurar um Arquivo ZIP:

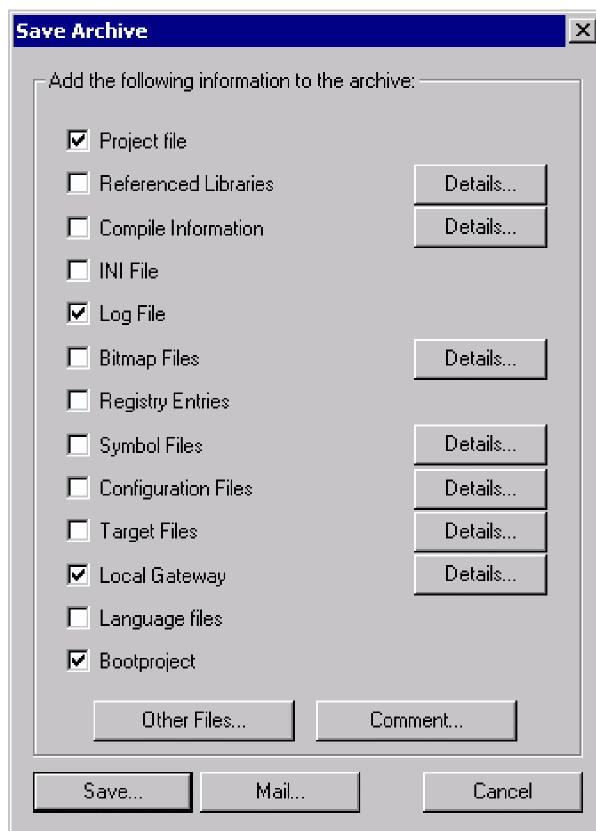


Figura 4-23. Caixa de diálogo para configurar um Arquivo ZIP

Aqui pode-se definir quais categorias de arquivo devem ser adicionadas ao arquivo ZIP: selecione ou desmarque uma categoria ativando/desativando a caixa de seleção correspondente. Para fazer isso, dê um único clique na caixa de seleção ou um clique duplo no nome da categoria. Se uma categoria está marcada com um símbolo ☑, todos os arquivos dessa categoria serão adicionados ao arquivo ZIP; se ele estiver marcado com outro símbolo ☐, nenhum dos arquivos será adicionado. Para selecionar arquivos individuais de uma categoria, pressione o botão **Detalhes**. O diálogo Detalhes será aberto com uma lista de arquivos disponíveis.

Caixa de diálogo para seleção detalhada de arquivos para o ZIP:

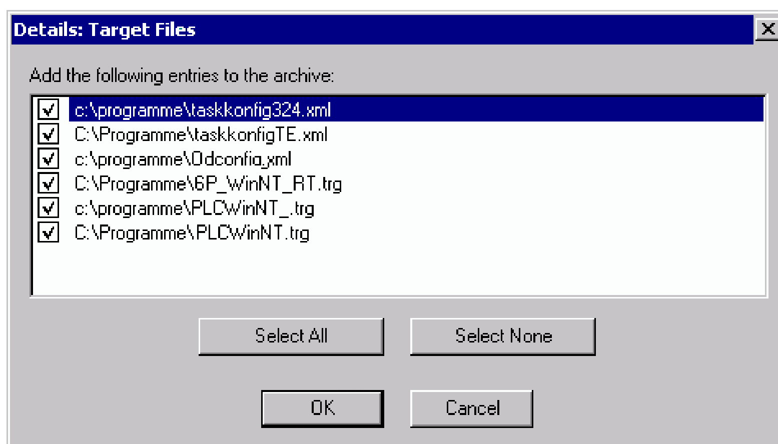


Figura 4-24. Caixa de diálogo para seleção detalhada de arquivos para o ZIP

O diálogo mostra uma lista de todos os arquivos que estão disponíveis para a categoria, onde todos os arquivos são selecionados automaticamente. Uma exceção é a categoria 'Arquivos de dispositivo', onde somente os arquivos que forem selecionados são relevantes para o sistema de dispositivo configurado atualmente. Para modificar a seleção ative e desative os arquivos desejados. Com o botão **Selecionar todos**, todos os arquivos da lista serão selecionados; com **Selecionar nenhum**, nenhum deles será. Um arquivo único pode ser selecionado/desmarcado por um clique no mouse na caixa de seleção, por um clique duplo na entrada da lista, ou também, pressionando a barra de espaços quando a entrada da lista é marcada.

Feche a caixa de diálogo Detalhes com **Salvar** para armazenar as novas configurações.

Na caixa de diálogo principal, a caixa de seleção das categorias, para as quais nem todos os arquivos foram selecionados, aparecerá com uma cor de plano de fundo cinza .

As seguintes categorias de arquivo estão disponíveis. A coluna à direita da tabela mostra quais arquivos podem ser adicionados ao arquivo compactado:

Categoria	**Arquivos**
Arquivo de projeto	nomedoprojeto.pro (o arquivo de projeto do MasterTool IEC)
Bibliotecas de referência	*.lib *.obj, *.hex (bibliotecas e, se disponível, os arquivos de objeto e hex correspondentes
Informação de compilação	*.ci (informações de compilação), *.ri (informações de download/referência) <temp>.* (arquivos de compilação temporária e de download) também para simulação
Arquivo INI	MToolIEC.ini
Arquivo Log	*.log (arquivo de log do projeto)
Arquivos de bitmap	*.bmp (bitmaps para POUs de projeto e visualizações)
Entradas de registro	Entradas de registro para Automation Alliance, Gateway e SPS; as seguintes sub-árvores serão compactadas: HKEY_LOCAL_MACHINE\SOFTWARE\3S-Smart Software Solutions HKEY_LOCAL_MACHINE\SOFTWARE\AutomationAlliance"
Arquivos de símbolos	*.sdb, *.sym (informação de símbolos)
Arquivos de configuração	Arquivos usados para a configuração do CP (arquivos de configuração, arquivos de dispositivo, ícones etc.): *.cfg por exemplo, * .CON, * .EDS *.dib, *.ico....
Arquivos de dispositivo	*.trg (arquivos de dispositivo no formato binário para todos os dispositivos instalados) *.txt (arquivos de dispositivo para os dispositivos instalados no formato de texto, se disponível)
Gateway local	Gateway.exe, GatewayDDE.exe, GClient.dll, GDrvBase.dll, GDrvStd.dll, GHandle.dll, GSymbol.dll, GUtil.dll, + DLLs no diretório do gateway, se disponível
Arquivos de idioma	Arquivos de idioma usados para visualização (*.vis, *.xml)
Projeto de Inicialização	Arquivos de projetos de inicialização <nomedoprojeto>.prg,

	<nomedoprojeto>.chk e arquivos do projeto de inicialização específicos do dispositivo

Tabela 4-2. Categorias de arquivo

Para adicionar outros arquivos ao ZIP, pressione o botão **Outros Arquivos**. Será aberta a caixa de diálogo 'Outros arquivos', na qual pode-se configurar uma lista de arquivos desejados.

Pressione o botão Adicionar para abrir a caixa de diálogo padrão para abrir um arquivo, onde pode-se procurá-lo. Escolha um e confirme com Abrir. Ele será adicionado à lista via diálogo 'Outros arquivos'. Repita isso para cada um deles que se deseja adicionar. Para excluir entradas da lista, pressione o botão Remover. Quando a lista de arquivos selecionados estiver como deseja-se, feche a caixa de diálogo com OK.

Caixa de diálogo para adicionar outros arquivos para o arquivo ZIP:

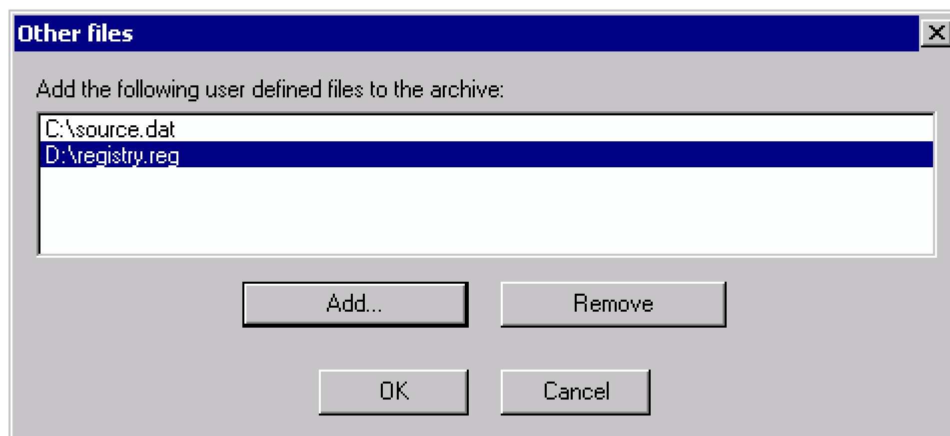


Figura 4-25. Caixa de diálogo para adicionar outros arquivos para o arquivo ZIP

Para adicionar um arquivo “Leia-me” no arquivo ZIP, pressione o botão **Comentário**. Um editor de texto será aberto, onde pode-se inserir qualquer texto. Ao fechar a caixa de diálogo com OK, durante a criação de arquivo ZIP um arquivo **leiame.txt** será adicionado. Além dos comentários inseridos, ele conterá informações sobre a data de criação e versão do MasterTool IEC.

Se todas as seleções desejadas tiverem sido feitas, na caixa de diálogo principal pressione:

Salvar... para criar e salvar o arquivo ZIP: a caixa de diálogo padrão para salvar um arquivo será aberta e poder-se-á inserir o caminho, onde o ZIP deve ser armazenado. O arquivo zip, por padrão, é chamado <nome do projeto>.zip. Confirme com Salvar para começar a criá-lo. Durante a criação o status do andamento atual é exibido e os passos subseqüentes são listados na janela de mensagens. Também será exibida uma mensagem se algum arquivo não pôde ser encontrado.

Enviar… para criar um arquivo temporário ZIP e para gerar automaticamente um e-mail vazio que contém o ZIP como um anexo. Esse recurso só funciona se o MAPI (Messaging Application Programming Interface) tiver sido instalado corretamente no sistema; caso contrário, será gerada uma mensagem de erro. Durante a instalação do e-mail, o status de progresso é exibido e as etapas da ação são listadas na janela de mensagens. O arquivo ZIP temporário será removido automaticamente após a conclusão da ação.

Cancelar ...para cancelar a ação; nenhum arquivo ZIP será gerado.

Menu ‘Arquivo’ ‘Imprimir’

Atalho: <Ctrl>+<P>

O conteúdo da janela ativa é impresso com este comando.

Depois que o comando for escolhido, a caixa de diálogo 'Imprimir' será exibida. Escolha a opção desejada ou configure a impressora e, em seguida, clique em **OK**. A janela ativa é impressa. A saída de cores está disponível em todos os editores.

Caixa de diálogo Imprimir

Pode-se determinar o **número de cópias** e imprimir a versão para um arquivo.

Com o botão '**Propriedades**', abre-se a caixa de diálogo para configurar a impressora.

Pode-se determinar o leiaute de impressão com o comando 'Arquivo' 'Configuração de impressão'.

Durante a impressão a caixa de diálogo mostra o número de páginas já impressas. Ao fechar esta caixa de diálogo, então, a impressão pára após a próxima página.

Para documentar o projeto inteiro, use o comando 'Projeto' 'Documentar'.

Para criar uma moldura de documento para o seu projeto, no qual pode-se armazenar comentários sobre todas as variáveis usadas nele, abra uma lista de variáveis globais e use o comando **'Extras' 'Criar arquivo de documentação'**.

Caixa de diálogo de impressão:

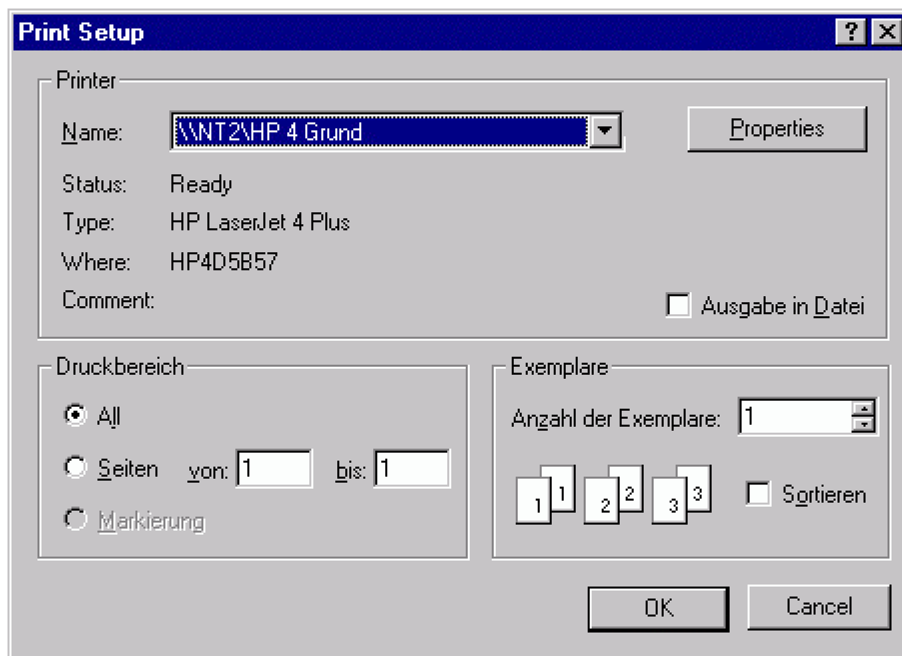


Figura 4-26. Caixa de diálogo de impressão

Menu 'Arquivo' 'Configurações de impressão'

Com este comando pode-se determinar o leiaute das páginas impressas. A seguinte caixa de diálogo é aberta.

Caixa de diálogo leiaute de página:

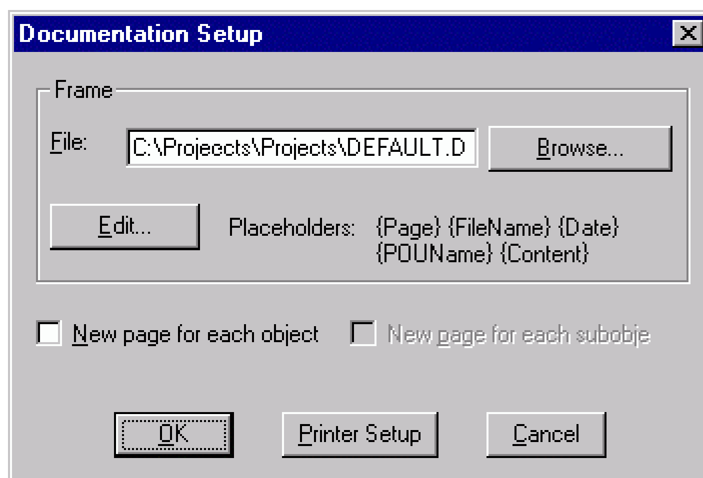


Figura 4-27. Caixa de diálogo leiaute de página

No campo '**Arquivo**' pode-se inserir o nome do arquivo com a extensão ".dfr", no qual o leiaute da página deve ser salvo. O dispositivo padrão para as configurações é o arquivo DEFAULT.DFR.

Para alterar um leiaute existente, procure a árvore de diretório para localizar o arquivo desejado, com o botão '**Procurar**'.

Também pode-se escolher se deseja-se iniciar uma **nova página para cada objeto** e **para cada sub-objeto**. Use o botão 'Configuração de impressora' para abrir a configuração da impressora.

Ao clicar no botão '**Editar**', o quadro para configurar o leiaute da página aparecerá. Aqui pode-se determinar os números de página, data, nome do arquivo e da POU e também posicionar gráficos na página e a área de texto em que a documentação deve ser impressa.

Janela para colar os espaços reservados no leiaute de página:

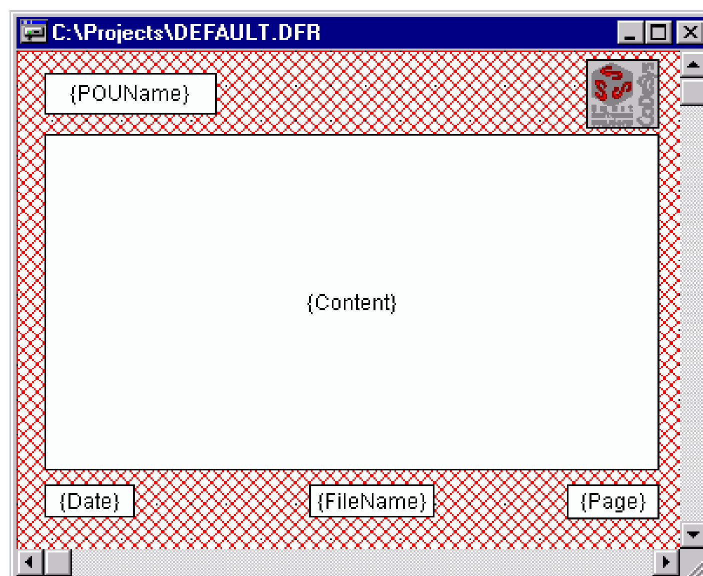


Figura 4-28. Janela para colar os espaços reservados no leiaute de página

Com o item de menu **'Inserir' 'Espaço reservado'** e a subseqüente seleção entre os cinco espaços reservados (**página, nome da POU, nome do arquivo, data e conteúdo**), insira no leiaute o chamado espaço reservado, arrastando um retângulo no leiaute enquanto pressiona o botão esquerdo do mouse. Na impressão, eles são substituídos da seguinte maneira:

Comando	Espaço reservado	Efeito
Página	{PAGE}	Aqui o número de páginas atuais aparece na impressão
Nome da POU	{POU Name}	Aqui o nome atual da POU é exibido.

Nome de arquivo	{File Name}	Aqui o nome do projeto é exibido.
Data	{Date}	Aqui a data atual é exibida.
Conteúdo	{Content}	Aqui o conteúdo da POU exibido.

Tabela 4-3. Substituição dos espaços reservados na impressão

Além disso, com **'Inserir' 'Bitmap'** pode-se inserir um gráfico de bitmap (por exemplo, um logotipo da empresa) na página. Depois de selecionar o elemento gráfico, um retângulo também deve ser desenhado aqui no leiaute usando o mouse. Outros elementos de visualização também podem ser inseridos (consultar Visualizações).

Se o modelo foi alterado, então o MasterTool IEC perguntará, quando a janela for fechada, se essas alterações devem ser salvas ou não.

Dica: para saber o formato de página que é válido para impressões, defina o leiaute conforme descrito acima e ative a opção 'Mostrar margens da área de impressão' nas opções do projeto, categoria Ambiente de trabalho.

Menu ‘Arquivo’ ‘Sair’

Atalho: <Alt> + <F4>

Com este comando pode-se sair do MasterTool IEC.

Se um projeto está aberto, então, ele será fechado conforme descrito em 'Arquivo' 'Salvar'.

Menu ‘Projeto’ ‘Compilar’

Atalho<F11>

O projeto é compilado usando 'Projeto' 'Compilar'. O processo de compilação é basicamente incremental, isto é, apenas as POUs alteradas são compiladas novamente. As informações necessárias sobre a última compilação são armazenadas em um arquivo *.ci quando o projeto é salvo. Uma compilação não-incremental também pode ser obtida se o comando 'Projeto' 'Limpar tudo' é executado pela primeira vez.

Em sistemas de dispositivo que suportam a 'Alteração Online', todas as POUs serão carregadas para o controlador no próximo download e serão marcadas com uma seta azul no Organizador de Objetos após a compilação.

O processo de compilação que é realizado com 'Projeto' 'Compilar' ocorre automaticamente se o controlador for conectado por meio de 'Comunicação' 'Login'.

Consultar um diagrama que mostra as relações entre Projeto-Compilação, Projeto-Download, Alteração Online e Login no sistema do dispositivo no capítulo 4: Comunicação, Login.

Durante a compilação é aberta uma janela de mensagens, a qual mostra o andamento do processo, os erros e avisos que podem ocorrer, bem como informações nos índices das POU usadas e espaço de memória (número e porcentagem). Erros e avisos são marcados com números. Usando F1 obtém-se mais informações sobre o erro atual selecionado.

Consultar a listagem de todas as mensagens de erro e avisos disponíveis.

Exemplo de mensagens de erro e informações de compilação na janela de mensagens de um projeto:

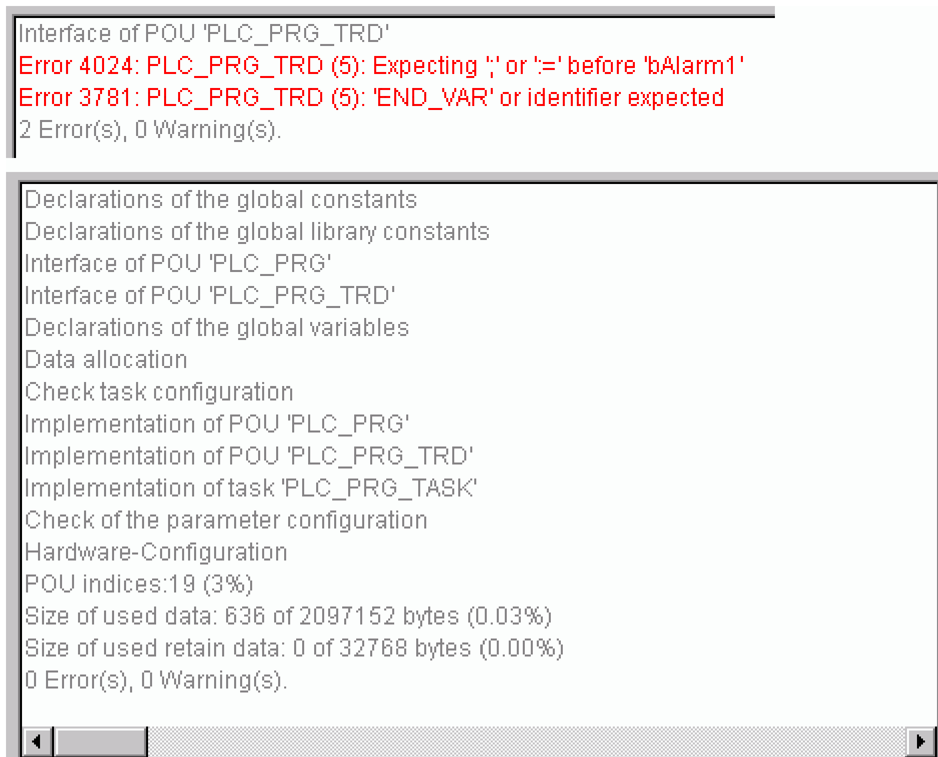


Figura 4-29. Mensagens de erro e informações de compilação

Se a opção **Salvar antes da compilação** está selecionada nas opções da caixa de diálogo categoria 'Carregar & Salvar', o projeto é armazenado antes da compilação.

Um ou diversos objetos selecionados no Organizador de Objetos podem ser excluídos da compilação pelo comando 'Excluir da compilação', o qual está disponível no menu de contexto, ou por meio de uma configuração apropriada ('Excluir objetos') em Opções para Compilação.

Nota: referências cruzadas são criadas durante a compilação e são armazenadas com as informações de compilação. Para poder usar o comando 'Mostrar árvore de chamadas' ou para obter resultados atualizados com os comandos 'Mostrar referência cruzada', 'Variáveis não usadas', 'Sobreposição de áreas de memória', 'Acesso concorrente', 'Múltiplos acessos de escrita na saída' (menu 'Projeto' 'Verificar'), compile novamente o projeto após qualquer alteração.

Menu ‘Projeto’ ‘Compilar tudo’

Com 'Projeto' 'Compilar tudo', ao contrário da compilação incremental ('Projeto' 'Compilar'), o projeto é totalmente re-compilado. Informações sobre download não são descartadas. Considere a possibilidade de excluir objetos da compilação.

Consultar um diagrama que mostra as relações entre Projeto-Compilação, Projeto-Download, Alteração Online e Login no sistema do dispositivo no capítulo 4: Comunicação, Login.

Menu ‘Projeto’ ‘Limpar tudo’

Com esse comando, todas as informações do último download e da última compilação serão excluídas.

Depois que o comando é selecionado, uma caixa de diálogo é aberta informando que não é mais possível realizar o Login sem um novo download (Alteração online). Nesse momento o comando pode ser cancelado ou confirmado.

Nota: depois de um 'Limpar tudo', somente será possível realizar uma Alteração Online no projeto do CP se o arquivo *.ri com as informações do último download do projeto foram renomeadas primeiro, ou então salvas fora do diretório do projeto (consultar 'Carregar informações de download').

Consultar um diagrama que mostra as relações entre Projeto-Compilação, Projeto-Download, Alteração Online e Login no sistema do dispositivo no capítulo 4: Comunicação, Login.

Menu 'Projeto' 'Carregar informações de download'

Com este comando as informações sobre download pertencentes ao projeto podem ser recarregadas. Depois de escolher o comando, a caixa de diálogo padrão 'Abrir arquivo', é aberta.

As informações de download são salvas automaticamente a cada download e, dependendo do sistema do dispositivo, também serão salvas a cada criação offline de um projeto de inicialização para um arquivo denominado **<nome do projeto> <identificador do dispositivo>.**ri e o qual é colocado para o diretório do projeto. Esse arquivo é recarregado cada vez que o projeto for reaberto e, no login, ele é usado para verificar o código de quais POUs foram alterados. Somente essas POUs, serão carregadas para o CP durante o procedimento de alteração online. Dessa forma, o arquivo *.ri é um pré-requisito para uma Alteração Online.

Considerar: se o arquivo *.ri no diretório do projeto é excluído pelo comando 'Projeto' 'Limpar tudo' somente pode ser recarregada a informação de download se o arquivo *.ri foi armazenado em outro diretório também. Ver capítulo 4: Comunicação, Login.

Consultar um diagrama que mostra as relações entre Projeto-Compilação, Projeto-Download, Alteração Online e Login no sistema do dispositivo no capítulo 4: Comunicação, Login.

Menu 'Projeto' 'Traduzir para outro idioma'

Este item de menu é usado para traduzir o arquivo do projeto atual em outro idioma. Isso é realizado pela leitura em um arquivo de tradução, o qual foi gerado a partir do projeto e externamente aprimorado no idioma desejado com a ajuda de um editor de texto. O projeto pode ser apenas exibido ou realmente ser convertido em uma das versões de idioma geradas.

Os seguintes subitens do menu estão presentes:

- Criar arquivo de tradução
- Traduzir projeto
- Mostrar projeto traduzido
- Alternar tradução

Ver também: 'Edição do arquivo de tradução'

Criar Arquivo de Tradução

Este comando no menu 'Projeto' 'Traduzir para outro idioma' leva à caixa de diálogo 'Criar arquivo de tradução'.

No campo **Arquivo de tradução**, digite um caminho que mostra onde o arquivo deve ser armazenado. A extensão padrão do arquivo é *.tlt; esse é um arquivo de texto. Também pode ser usada a extensão *.txt , que é recomendada caso desejar-se trabalhar em arquivo do Excel ou Word, porque nesse caso, os dados são organizados em formato de tabela.

Para processar um arquivo de tradução já existente, forneça o caminho desse arquivo ou use o botão **Procurar** para obter a caixa de diálogo padrão de seleção de arquivo do Windows.

As seguintes informações a partir do projeto podem ser, opcionalmente, transmitidas para o arquivo de tradução que está sendo modificado ou criado, para que elas estejam disponíveis para a tradução: **Nomes** (nomes, por exemplo, os títulos das 'POUs' no Organizador de Objetos), **Identificadores**, **Strings**, **Comentários**, **Textos de Visualização**, **Textos de Alarme**. Além dessas, **Informações de posição** para esses elementos do projeto também podem ser ser transferidas.

Caixa de diálogo para criar um arquivo de tradução:

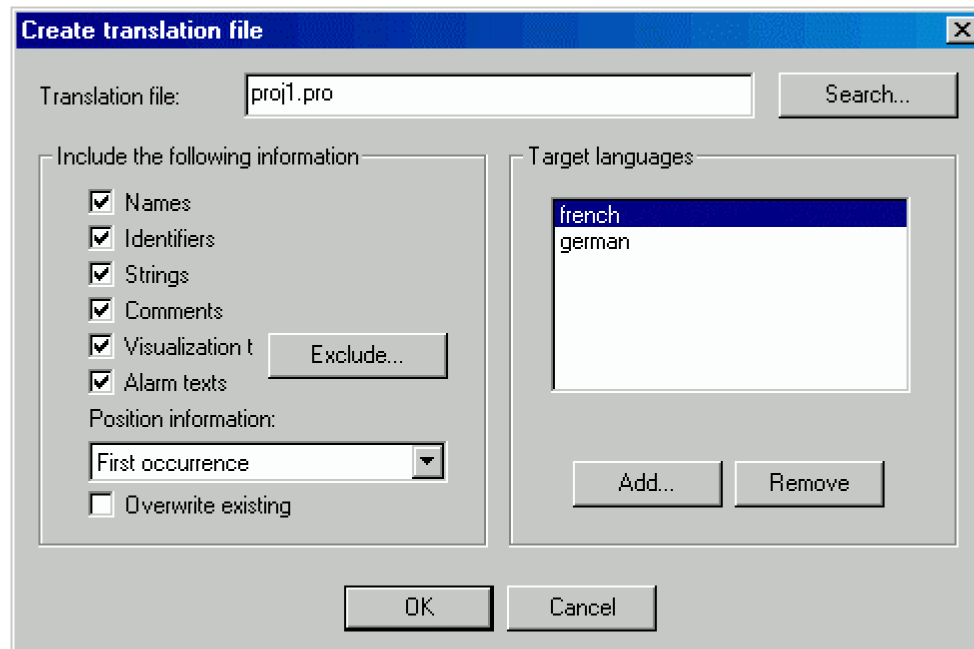


Figura 4-30. Caixa de diálogo para criar um arquivo de tradução

Se as opções correspondentes são verificadas, as informações a partir do projeto atual serão exportadas como símbolos de idioma em um arquivo de tradução recém-criado ou adicionadas a um já existente. Se a opção respectiva não estiver selecionada, informações pertencentes à categoria pertinente, independentemente de qual projeto vieram, serão excluídas do arquivo de tradução.

<u>Textos de Visualizações</u>

A opção 'Visu. (txt)' diz respeito apenas à 'Texto' e a opção ‘Texto tooltip’ refere-se a um elemento de visualização. Considere os seguintes itens ao usar um arquivo de tradução para textos de visualizações...

Um arquivo de tradução *.tlt- ou *.txt só pode ser usado com MasterTool IEC ou MasterTool IEC HMI, porém nunca com Visualização do Dispositivo ou Visualização Web. Talvez seja melhor usar um arquivo de idioma de visualização especial *. vis.

Chaveamento para outro idioma só é possível no modo online. Isso significa que os textos da visualização não serão traduzidos pelo comando 'Traduzir para outro idioma'. Uma alteração de idioma só pode ocorrer no modo online se o idioma correspondente for inserido na caixa de diálogo 'Extras' 'Configurações'.

Se os arquivos *.tlt ou *.txt devem ser usados para textos de visualização ('Texto' e 'Texto para Tooltip'), eles devem ser colocados entre parênteses por dois símbolos de "#" na caixa de diálogo de configuração do elemento de visualização (por exemplo, #texto#), para que possam ser transferidos para o arquivo de tradução.

Referente ao chaveamento de idioma nas telas, consultar o Manual de Utilização MasterTool IEC – Visualização.

Informação de posição: descreve - através das especificações do caminho do arquivo, POU e linha - a posição do símbolo do idioma disponibilizado para tradução. Três opções estão disponíveis para seleção:

'Nenhuma':	Nenhuma informação de posição é gerada.
'Primeira ocorrência':	A posição na qual o elemento aparece primeiro é adicionada ao arquivo de tradução.
'Todas':	Todas as posições nas quais o elemento correspondente aparece são especificadas.

Tabela 4-4. Opções para Informação de posição

Se estiver para ser editado um arquivo de tradução criado anteriormente que contenha mais informações de posição do que o arquivo atualmente selecionado, ele será truncado ou excluído, não importando a partir de qual projeto foi gerado.

Nota: um máximo de 64 especificações de posição será gerado por elemento (símbolo de idioma), mesmo se o usuário tiver selecionado "Todas" em "Informações de posição" no diálogo 'Criar arquivo de tradução'.

Substituir existente: informações de posição existentes no arquivo de tradução, que está atualmente sendo processado, serão sobrescritas, independentemente do projeto a partir do qual foi gerado.

Idiomas destino: esta lista contém identificadores para todos os idiomas que estão contidos no arquivo de tradução, bem como aqueles a serem adicionados após a finalização do diálogo 'Criar arquivo de tradução'.

O botão **Excluir** abre a caixa de diálogo 'Excluir bibliotecas'. Ali, podem ser selecionadas as bibliotecas incluídas no projeto, cujas informações de identificação ainda não foram transferidas para o arquivo de tradução. Para fazer isto, selecione a entrada correspondente na tabela, no campo **Bibliotecas incluídas** (à esquerda) e coloque-a no campo **Bibliotecas excluídas** (à direita), usando o botão **Adicionar**. Da mesma forma, as entradas já colocadas lá podem ser removidas usando o botão **Remover**. OK confirma a configuração e fecha a caixa de diálogo.

Caixa de diálogo para excluir informações de biblioteca no arquivo de tradução:

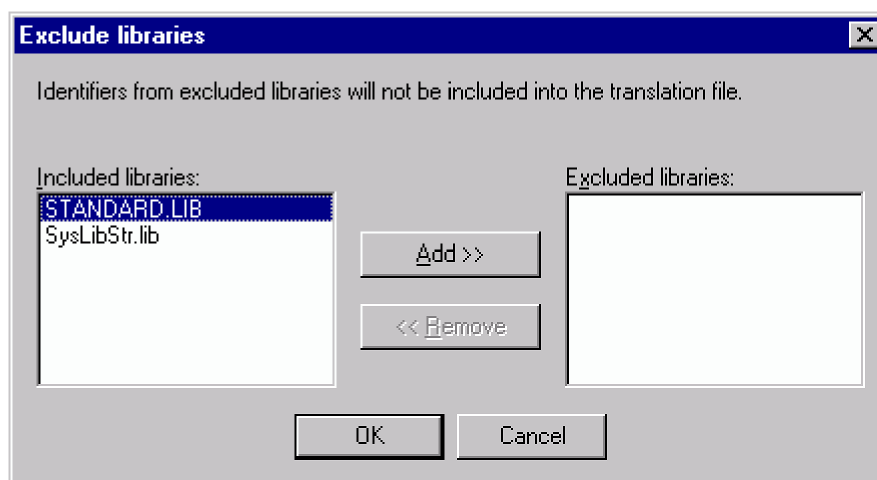


Figura 4-31. Caixa de diálogo para excluir informações de biblioteca no arquivo de tradução

O botão **Adicionar** abre a caixa de diálogo 'Adicionar idioma do dispositivo'.

Caixa de diálogo para adicionar um idioma (Projeto, Traduzir para outro idioma):



Figura 4-32. Caixa de diálogo para adicionar um idioma destino

Um identificador de idioma deve ser inserido no campo de edição; ele não pode ter espaço ou caractere que tenha trema (ä, ö, ü), tanto no início quanto no final.

OK fecha a caixa de diálogo 'Adicionar idioma do dispositivo' e o novo idioma aparece na lista de idiomas.

O botão **Remover** remove uma entrada selecionada da lista.

Também pode-se confirmar a caixa de diálogo 'Criar arquivo de tradução' com um **OK**, para gerar um arquivo de tradução.

Se já existe um arquivo de tradução com o mesmo nome, aparecerá a seguinte mensagem de confirmação, a ser respondida Sim ou Não:

"O arquivo de tradução especificado já existe. Agora ele será alterado e será criada uma cópia de backup do arquivo existente. Você deseja continuar?"

Não retorna (sem ação) para a caixa de diálogo 'Criar arquivo de tradução'. Se **Sim** for selecionado, uma cópia do arquivo de tradução existente com o nome de arquivo "Backup_of_ <arquivo de tradução>.xlt" será criada no mesmo diretório e o arquivo de tradução correspondente será modificado de acordo com as opções que foram inseridas.

O seguinte ocorre quando um arquivo de tradução é gerado:

Para cada novo idioma destino, um espaço reservado ("##TODO") é gerado para cada símbolo de idioma a ser exibido (consultar 'Editando o arquivo de tradução' para saber como trabalhar no arquivo de tradução).

Se um arquivo de tradução existente for processado, as entradas no arquivo de idiomas que aparecem no arquivo de tradução, mas não na lista de idiomas destino, serão excluídas, independentemente do projeto a partir do qual foram geradas.

Edição do arquivo de tradução

O arquivo de tradução deve ser aberto e salvo como um arquivo de texto. Os sinais ## marcam palavras-chave. Os espaços reservados (##TODO) no arquivo podem ser substituídos por traduções válidas. Para cada símbolo de idioma é gerado um parágrafo que começa e termina com um identificador de tipo. Por exemplo ## NAME_ITEM e ## END_NAME_ITEM inclui uma seção para o nome de um objeto como usado no Organizador de Objetos (seções de marcas COMMENT_ITEM para comentários, IDENTIFIER_ITEM para identificadores, STRING_ITEM para seqüências de caracteres e VISUALTEXT_ITEM para textos de visualização).

Consultar o seguinte exemplo de tradução de um parágrafo do arquivo que lida com o nome de uma das POUs do projeto ST_Visu. Os idiomas do dispositivo devem ser Inglês (USA) e Francês. Nesse exemplo, as informações de posição do elemento do projeto que deveria ser traduzido foram adicionadas:

Antes da tradução:

```
##NAME_ITEM
[D:\MASTERTOOLIEC\PROJECTS\BSPDT_22.PRO::ST_VISUALISIERUNG::0]
ST_VISUALISIERUNG
##ENGLISH :: ##TODO
```

##FRENCH :: ##TODO

##END_NAME_ITEM

Após a tradução: o ##TODOs foram substituídos pela palavra francesa e inglesa para 'Visualisierung':

##NAME_ITEM

[D:\ MASTERTOOLIEC\PROJECTS\BSPDT_22.PRO::ST_VISUALISIERUNG::0]

ST_VISUALISIERUNG

##ENGLISH :: ST_VISUALIZATION

##FRENCH :: ST_VISU

##END_NAME_ITEM

Por favor, certifique-se de que os identificadores e os nomes traduzidos permaneçam válidos em relação ao padrão e que as seqüências de caracteres e comentários estejam em colchetes corretos. Exemplo: para um comentário (## COMMENT_ITEM) que é representada com "(* Kommentar 1 *) "no arquivo de conversão, o "##TODO" em "##English" deve ser substituído por um "(* Comment 1 *)". Para uma seqüência de caracteres (## STRING_ITEM) representada com "zeichenfolge1" o "##TODO" deve ser substituído por "string1".

Dica: as seguintes partes de um arquivo de tradução não devem ser modificadas sem conhecimento detalhado: bloco de idioma, bloco sinalizador, informações de posição, textos originais.

Traduzir o projeto (para outro idioma)

Este comando no menu 'Projeto' 'Traduzir este projeto' abre a caixa de diálogo 'Traduzir projeto para outro idioma'.

Caixa de diálogo para traduzir o projeto para outro idioma:



Figura 4-33. Caixa de diálogo para traduzir o projeto para outro idioma

O projeto atual pode ser traduzido para outro idioma se um arquivo de tradução apropriado é usado.

Nota: para salvar a versão do projeto no idioma no qual ele foi originalmente criado, salve uma cópia do projeto com um nome diferente antes de traduzir. **O processo de tradução não pode ser desfeito.**

Considere, nesse contexto, a possibilidade apenas de exibir o projeto em outro idioma (nessa versão de exibição, porém não editável).

No campo de **Tradução de arquivo**, forneça o caminho para o **arquivo de tradução** a ser usado. Pressionando **Localizar** pode-se acessar a caixa de diálogo padrão de seleção de arquivo do Windows.

O campo **Idioma do dispositivo** contém uma lista de identificadores de idioma inseridos no arquivo de tradução, a partir da qual pode-se selecionar o idioma do dispositivo desejado.

OK inicia a tradução do projeto atual para o idioma do dispositivo escolhido com a ajuda do arquivo de tradução especificado. Durante a tardução, uma caixa de diálogo de progresso é exibida, bem como mensagens de erro, se houver alguma. Após a conversão, a caixa de diálogo e todas as janelas abertas do editor do projeto são fechadas.

Cancelar fecha a caixa de diálogo sem modificações no projeto atual.

Se o arquivo de tradução contiver entradas erradas, uma mensagem de erro será exibida depois que OK for pressionado, dando o caminho do arquivo e a linha errada, por exemplo:

"[C:\PROGRAMS\MASTERTOOLIEC\PROJECTS\VISU.TLT (78)]; TRADUÇÃO DE TEXTO ESPERADA"

Nota: considere as especialidades para textos de visualizações; veja acima 'Projeto' 'Criar arquivo de tradução'.

'Mostrar projeto traduzido'

Se houver um arquivo de tradução disponível para o projeto, pode-se exibir uma das versões de idioma definidas ali, sem sobrescrever a versão de idioma original do projeto.

Compare esta possibilidade com a tradução “real" de um projeto que seria feita com o comando 'Traduzir projeto' e veja que esta implica em criar uma nova versão do projeto!

O comando 'Traduzir este projeto' no menu 'Projeto' 'Traduzir para outro idioma' abre a caixa de diálogo 'Traduzir projeto para outro idioma'.

Caixa de diálogo para exibir o projeto em outro idioma:

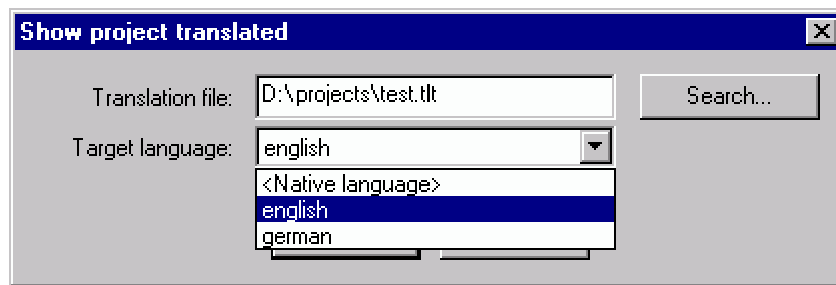


Figura 4-34. Caixa de diálogo para exibir o projeto em outro idioma

No campo **Arquivo de tradução**, insira o caminho do arquivo de tradução que deseja-se usar. Pode-se receber assistência para abrir um arquivo através da caixa de diálogo padrão, via **Procurar**.

No campo **Idioma do dispositivo** se encontra uma lista de seleção, que além da entrada "<Idioma nativo>", também oferece os identificadores de idioma que estão definidos pelo arquivo de tradução atual. O idioma original é aquele salvo atualmente no projeto (ele somente pode ser alterado com 'Projeto' 'Traduzir'). Escolha um dos idiomas disponíveis e confirme o diálogo com OK. Dessa maneira o projeto será exibido no idioma escolhido, **porém nesse modo de exibição não pode ser editado!**

Se desejar retornar para exibir o projeto em seu idioma original, use o comando 'Alternar tradução'.

Nota: considere as especialidades para textos em visualizações.

'Alternar tradução'

Se o projeto (não editável) for exibido em outro idioma pelo comando 'Mostrar projeto traduzido', pode-se agora alternar entre essa versão do idioma e a versão original (editável) usando o comando 'Alternar tradução' no menu 'Projeto' 'Traduzir para outro idioma)'.

Nota: considere as especialidades para textos em visualizações; veja acima 'Projeto' 'Criar arquivo de tradução'.

Menu ‘Projeto’ ‘Documentar’

Este comando permite imprimir a documentação do projeto inteiro.

Caixa de diálogo para documentação do projeto:

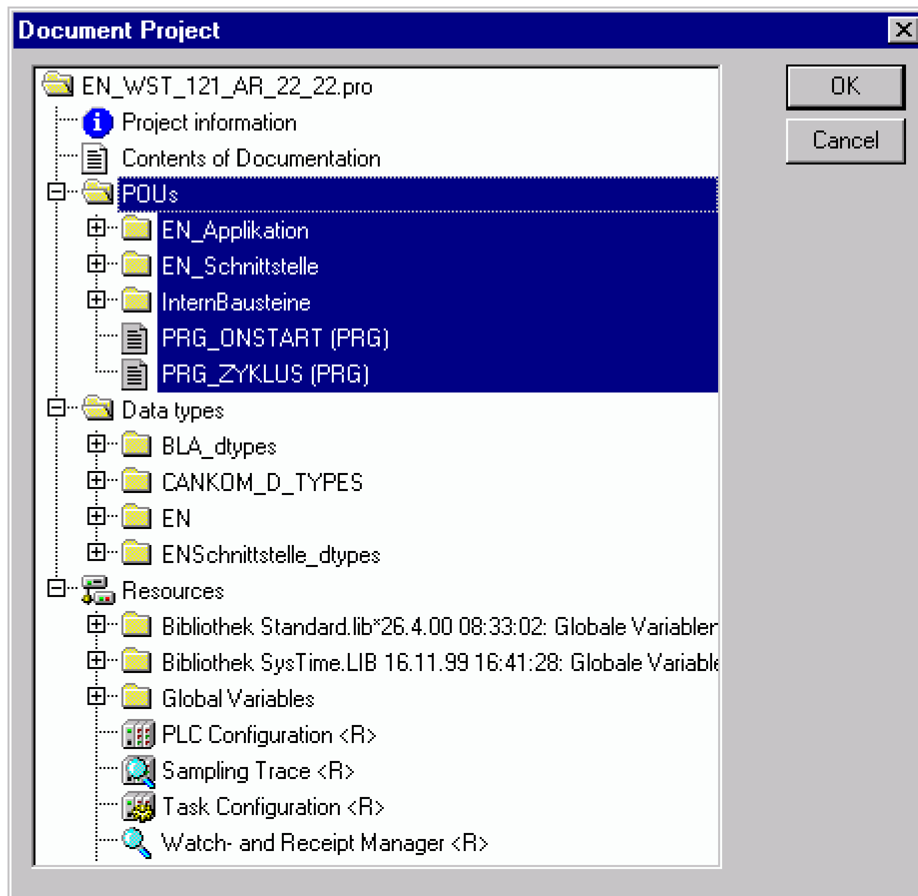


Figura 4-35. Caixa de diálogo para documentação do projeto

Os elementos de uma documentação completa são:

- As POUs;
- O conteúdo da documentação;
- Os tipos de dados;
- As visualizações;
- Os recursos (variáveis globais, variáveis de configuração e outras configurações e gerenciadores);
- As árvores de chamada de POUs e tipos de dados ;
- A lista de referência cruzada.

Para os dois últimos itens o projeto deve ter sido compilado sem erros.

Somente aquelas áreas na caixa de diálogo são impressas com realce em azul.

Para selecionar o projeto inteiro, selecione o nome do seu projeto na primeira linha.

Se, por outro lado, apenas deseja-se selecionar um único objeto, clique no objeto correspondente ou mova o retângulo pontilhado até o objeto desejado com a tecla de seta. Objetos que possuem um sinal de mais na frente de seus símbolos são objetos de organização que contêm outros objetos. Com um clique no sinal de mais, o objeto de organização é expandido e, com um clique no sinal de subtração, ele pode ser fechado novamente. Ao selecionar um objeto de organização, todos os objetos relevantes

são também selecionados. Pressionando a tecla <Shift> pode-se selecionar um grupo de objetos e pressionando a tecla <Ctrl> pode-se selecionar vários objetos individuais.

Depois da seleção, clique em **OK**. A caixa de diálogo 'Imprimir' será exibida. Pode-se determinar o leiaute das páginas a serem impressas com 'Arquivo' 'Configuração de impressão'.

Menu 'Projeto' 'Exportar'

Com o **MasterTool IEC**, os projetos podem ser exportados ou importados. Isso permite a troca de programas entre diferentes sistemas de programação IEC.

Existe um formato de troca padronizado para POUs em IL, ST e SFC (o formato de elementos comuns da IEC 61131-3). As POUs em LD e FBD e outros objetos do **MasterTool IEC** apresentam o seu próprio formato de arquivamento, desde que não haja nenhum formato de texto para isso na IEC 61131-3.

Os objetos selecionados são gravados em um arquivo ASCII.

POUs, tipos de dados, visualizações e recursos podem ser exportados. Além disso, também podem ser exportadas as entradas no gerenciador de bibliotecas, que são as informações de vinculação às bibliotecas (não as bibliotecas em si!).

Importante: importar novamente uma POU (em FBD ou LD) exportada resultará em erro se um comentário no editor gráfico contiver uma aspa simples ('), pois isso será interpretado como o início de uma string!

Depois da seleção na janela da caixa de diálogo (da mesma forma que com 'Projeto' 'Documentar'), pode-se decidir, se deseja exportar partes selecionadas para um arquivo ou se prefere exportar um arquivo para cada objeto em arquivos separados. Ative ou desative a opção **Um arquivo para cada objeto** e clique em **OK**. A caixa de diálogo para salvar arquivos aparece. Digite um nome de arquivo com a extensão".exp" em um diretório para os objetos exportados e, então, ele será salvo ali com o nome do arquivo <nomedoobjeto.exp>.

Menu 'Projeto' 'Importar'

Na caixa de diálogo para abrir arquivos, selecione o arquivo de importação desejado.

Os dados são importados para o projeto atual. Se um objeto com o mesmo nome já existe no projeto então, é exibida uma caixa de diálogo com a pergunta: "Você deseja substituí-lo?" Ao responder **Sim**, o objeto no projeto é substituído pelo objeto do arquivo de importação. Ao responder **Não**, os nomes dos objetos novos recebem um suplemento: um sublinhado e um dígito ("_0", "_1"..). Com **Sim, Todos** ou **Não, Todos** isso é realizado para todos os objetos ou somente para aquele.

Se as informações são importadas para o link com uma biblioteca, essa será carregada e acrescentada ao final da lista no gerenciador de bibliotecas. Se a biblioteca já foi carregada para o projeto, ela não será recarregada. Se, no entanto, o arquivo que está sendo importado mostra um tempo de armazenamento diferente para a biblioteca, o nome dessa é marcado com um "*" no gerenciador da biblioteca (ex.: standard.lib*30.3.99 11:30:14), semelhante ao carregamento de um projeto. Se a biblioteca não for encontrada, uma caixa de diálogo de informações será exibida: "Não é possível encontrar biblioteca {<caminho> \} <nome> <data> <hora>", da mesma forma que ocorre quando um projeto é carregado.

A importação é registrada na janela de mensagens.

Menu 'Projeto' 'Comparar'

Este comando é usado para comparar dois projetos ou para comparar a versão atual de um projeto com o que foi salvo pela última vez.

Nomenclatura da comparação de projetos:

Projeto Atual:	Projeto no qual se está trabalhando atualmente.
Projeto de referência:	Projeto que deve ser comparado com o projeto atual.
Modo comparação:	Nesse modo o projeto será exibido após o comando 'Projeto' 'Comparar' ter sido executado.
Unidade:	Menor unidade que pode ser comparada. Pode ser uma linha (editor de declaração, editor ST, editor IL), uma rede (editor FBD, editor LD) ou um elemento/POU (editor CFC, editor SFC).

Tabela 4-5. Nomenclatura da comparação de projetos

No modo comparação, o projeto atual e o projeto de referência serão apresentados em uma janela bipartida. Os nomes das POUs, para as quais foram encontradas diferenças, são marcados por cor. Para editores de POUs, também os conteúdo destes serão exibidos na maneira “vis-à-vis”. Os resultados e a maneira de apresentação no modo comparação dependem:

1. De quais filtros foram ativados para a execução da comparação, afetando a consideração de espaços em branco e comentários durante a comparação;

2. Se a modificação dentro de linhas, redes ou elementos é avaliada como uma inserção completamente nova de uma POU ou não.

A versão do projeto de referência pode ser aceita para diferenças únicas ou para todas marcadas igualmente. Aceitar significa que a versão do projeto de referência é adotada para o projeto atual.

Nota: no modo comparação (consultar barra de status: COMPARAR) o projeto não pode ser editado!

Executar comparação

Depois de executar o comando 'Projeto' 'Comparar', a caixa de diálogo para **Comparação de projeto** abre.

Caixa de diálogo para Comparação de projeto:

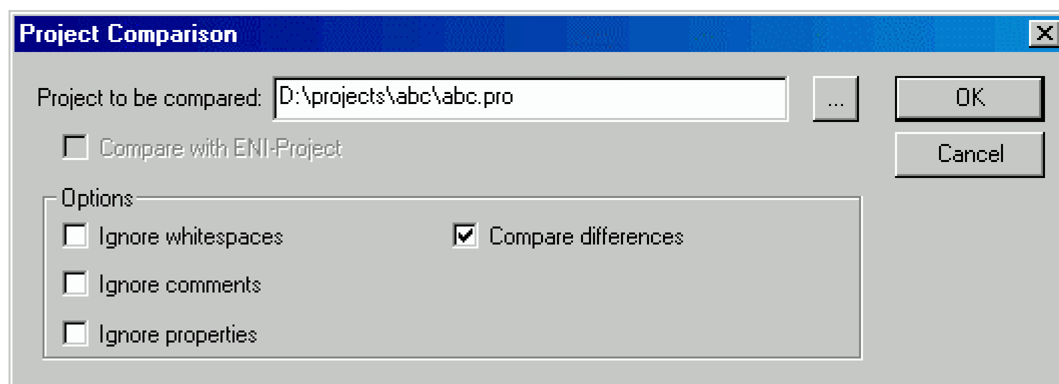


Figura 4-36. Caixa de diálogo para Comparação de projeto

Insira o caminho do projeto de referência no **Projeto para comparar**. Pressione o botão ▢ para usar a caixa de diálogo padrão para abrir um projeto. Ao inserir o nome do projeto atual, a versão atual dele vai ser comparada com a versão salva pela última vez.

Se o projeto estiver sob controle de fonte em uma base de dados de ENI, então, a versão local pode ser comparada com a versão atual encontrada na base de dados. Para isto, ative a opção **Comparar com projeto ENI**.

As seguintes opções relacionadas à comparação podem ser ativadas:

Ignorar espaços em branco: não será detectada nenhuma diferença que consista em números diferentes de espaços em branco.

Ignorar comentários: não será detectada nenhuma diferença nos comentários.

Ignorar propriedades: não será detectada nenhuma diferença nas propriedades do objeto.

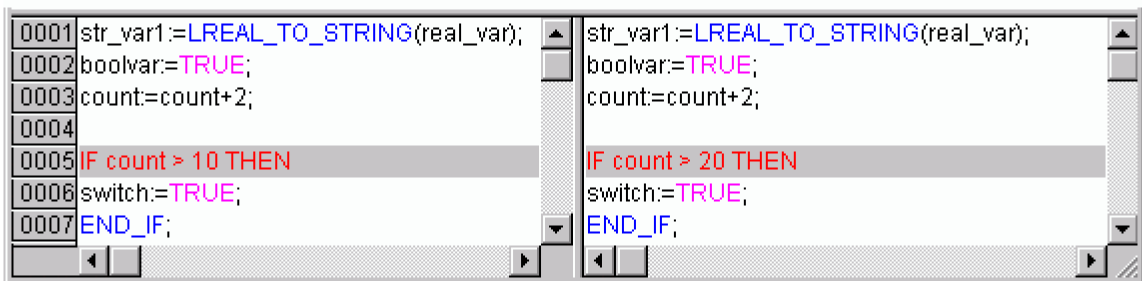
Comparar diferenças: no modo comparação, se uma linha, uma rede ou um elemento dentro de uma POU tiverem sido modificados, eles serão exibidos em uma janela bipartida diretamente oposta à versão do outro projeto (marcado como vermelho, ver abaixo). Se a opção estiver desativada, a linha correspondente será exibida no projeto de referência como 'excluída' e no projeto atual como 'inserida' (azul/verde, veja abaixo). Isso significa que ela não será exibida diretamente oposta à mesma linha no outro projeto.

Exemplo:

Linha 0005 foi modificada no projeto atual (lado esquerdo).

Exemplo para Diferenças:

Option 'Oppose differences' activated:



Option 'Oppose differences' not activated:

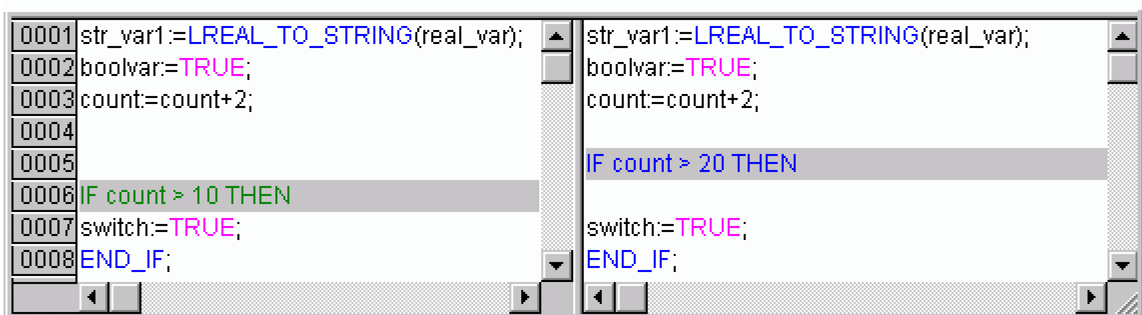


Figura 4-37. Exemplo para diferenças na comparação de projetos

Quando a caixa de diálogo 'Comparação de projeto' é fechada ao pressionar OK, a comparação será executada conforme as configurações.

Representação do resultado de comparação

Primeiramente, a árvore de estrutura do projeto - intitulada "Comparação de projetos"- será aberta para exibir os resultados da comparação. Aqui pode-se selecionar POUs específicas para ver detalhadamente as diferenças encontradas.

1. Visão geral do projeto no modo comparação

Depois que a comparação do projeto tiver sido executada, uma janela bipartida abre e mostra o projeto no modo comparação. Na barra de título aparecem os caminhos de projeto: "Comparação de projeto <caminho do projeto atual> - <caminho do projeto de referência>". O projeto atual é representado na metade esquerda da janela e o projeto de referência na direita. Cada árvore de estrutura mostra os nomes dos projetos na primeira posição, separada daquela que corresponde à estrutura do Organizador de Objetos.

POUs diferentes são marcadas na árvore de estrutura por uma sombra, uma cor específica e, eventualmente, por um texto adicional:

Vermelho: a unidade foi modificada e é exibida na cor vermelha em ambos os lados da janela.

Azul: unidade disponível somente no projeto de referência; uma lacuna será inserida no local correspondente na visão geral de estrutura do projeto atual.

Verde: unidade somente disponível no projeto atual; uma lacuna será inserida no local correspondente na visão geral da estrutura do projeto atual.

Preto: unidade para a qual nenhuma diferença foi detectada.

"(Propriedades alteradas)": este texto será anexado ao nome da POU na árvore de estrutura de projeto, se forem detectadas diferenças nas propriedades da POU.

"(Direitos de acesso alterados)": este texto será anexado ao nome da POU na árvore de estrutura de projeto, se forem detectadas diferenças nos direitos de acesso da POU.

Exemplo de projeto no modo comparação:

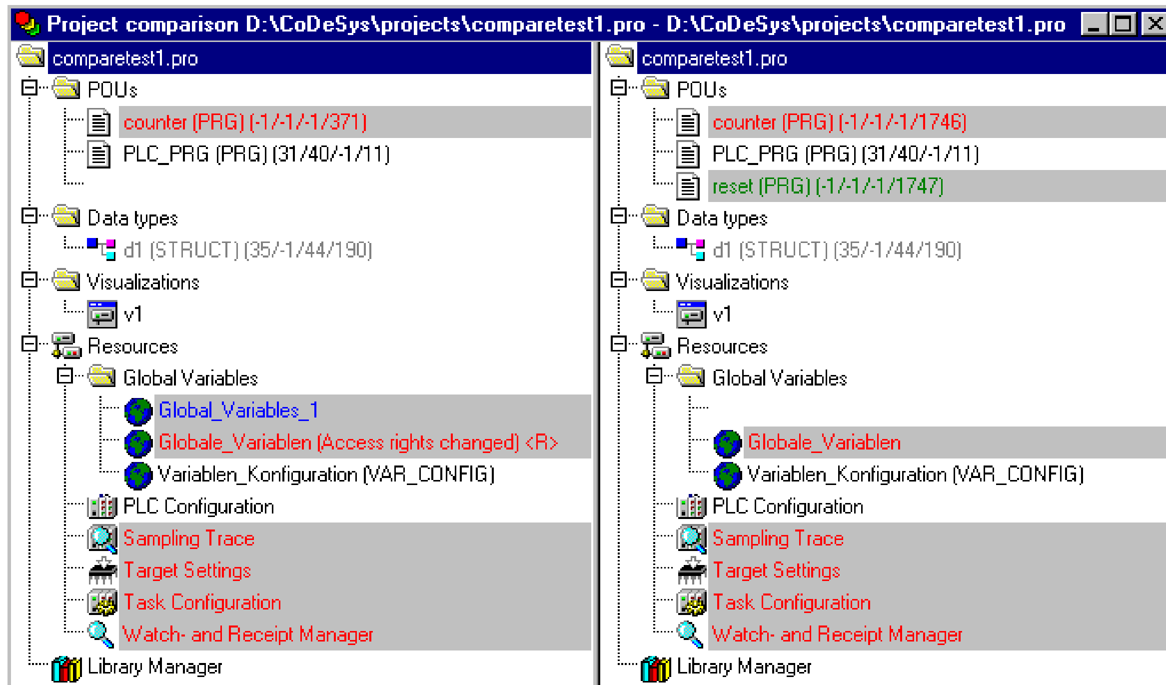


Figura 4-38. Projeto no modo comparação

2. Conteúdo da POU no modo comparação:

Na visão geral da estrutura, a POU é aberta através de um clique duplo na linha marcada em vermelho devido a uma modificação.

Se a POU for proveniente de um editor textual ou gráfico, ela será aberta em uma janela bipartida. O conteúdo do projeto de referência (lado direito) está do lado oposto ao projeto atual (lado esquerdo). A menor unidade que será considerada durante a comparação, é uma linha (editor de declaração, ST, IL), uma rede (FBD, LD) ou um elemento (CFC, SFC). A mesma coloração será usada, como descrito acima, para obter a visão geral sobre o projeto.

Exemplo da POU no modo comparação:

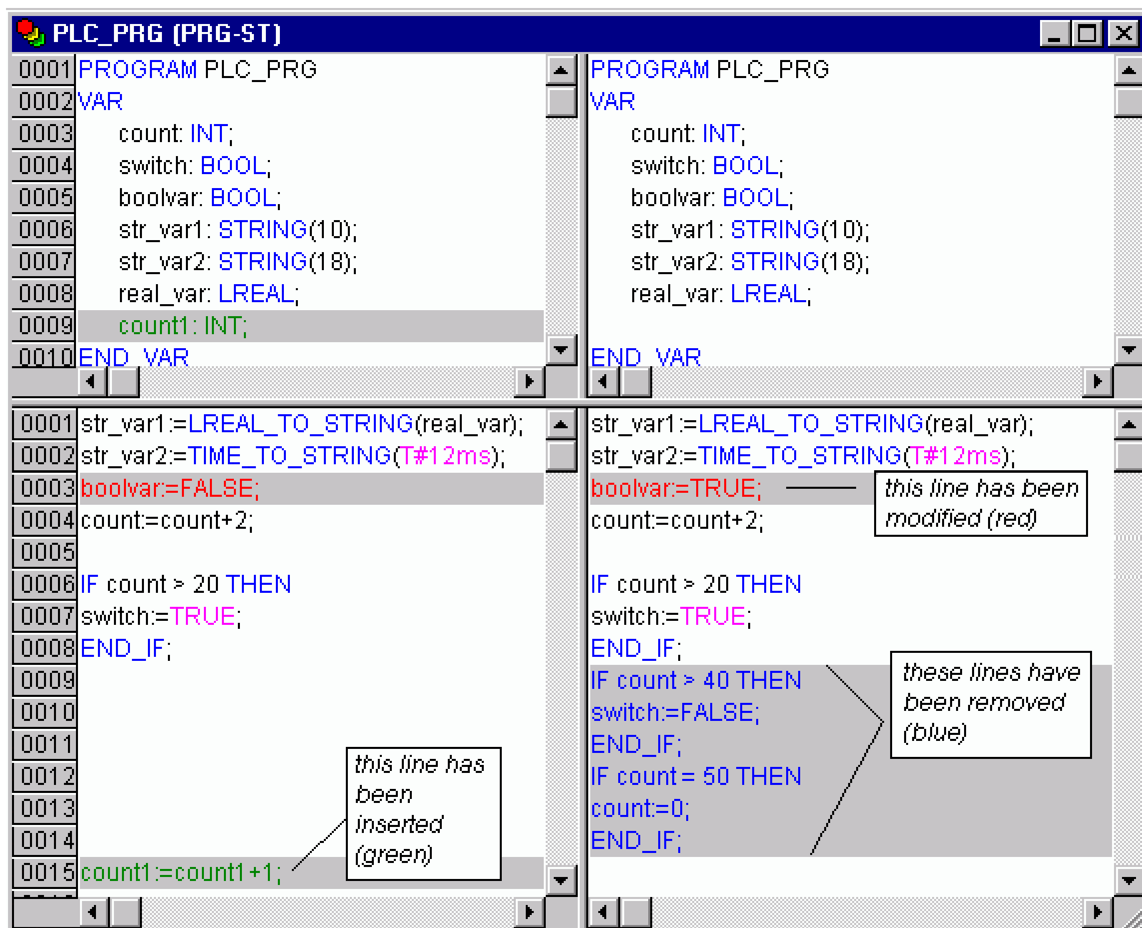


Figura 4-39. POU no Modo comparação

Se a POU referir-se a uma configuração de tarefa, a configurações do dispositivo ou outras que não as citadas acima, então, a versão da POU dos projetos atual e de referência podem ser abertas em janelas separadas através um clique duplo na respectiva linha na estrutura de projeto. Para essas POUs de projeto não serão exibidos detalhes adicionais das diferenças.

Trabalhando no modo comparação

Se o cursor estiver colocado em uma linha na janela bipartida, o que indica que há uma diferença, no menu 'Extras' ou no menu de contexto estará disponível uma seleção dos seguintes comandos, dependendo se estiver trabalhando em uma POU ou na visão geral do projeto.

Nota: a aceitação de diferentes partes do projeto (diferenças) ou propriedades de direito de acesso só serão possíveis a partir do projeto de referência e não no projeto atual. O contrário não vale.

'Extras' 'Próxima Diferença'

Atalho: <F7>

Este comando está disponível no modo comparação (consultar 'Projeto' 'Comparar').

O cursor vai para a próxima unidade, onde está indicada a diferença (linha na visão geral do projeto, linha/rede/elemento na POU).

'Extras' 'Diferença anterior'

Atalho: <Shift>+<F7>

Este comando está disponível no modo comparação (consultar 'Projeto' 'Comparar').

O cursor vai para a anterior, onde está indicada a diferença (linha na visão geral do projeto, linha/rede/elemento na POU).

'Extras' 'Aceitar alteração'

Atalho: <Espaço>

Este comando está disponível no modo comparação (consultar 'Projeto' 'Comparar').

Para todas as unidades coerentes com aquela, onde o cursor no momento está posicionado, e que tenham o mesmo tipo de diferença, (por exemplo, linhas subseqüentes), a versão do projeto de referência será aceita para o projeto atual (só é possível nessa direção!). As unidades correspondentes serão mostradas (com a cor correspondente) no lado esquerdo da janela.

Para aceitar as alterações de determinadas unidades, por favor, use 'Aceitar item alterado'.

Nota: a aceitação de diferentes partes do projeto (diferenças) ou propriedades de direito de acesso só serão possíveis a partir do projeto de referência do projeto atual. O contrário não vale.

'Extras' 'Aceitar item alterado'

Atalho: <Ctrl>+<barra de espaço>

Este comando está disponível no modo comparação (consultar 'Projeto' 'Comparar').

Somente as unidades únicas (linha, rede, elemento), onde o cursor está posicionado no momento, serão aceitas para a versão atual. As unidades correspondentes serão mostradas (com a cor correspondente) no lado esquerdo da janela.

Se for aceita a alteração de uma POU em vermelho (está nessa cor devido a uma alteração em seu conteúdo), então a cor dessa POU, no projeto atual, será substituída por amarelo.

As POUs que só estiverem disponíveis no projeto atual, somente por causa de 'Aceitar item alterado', também serão indicadas pela cor amarela. Da mesma forma, POUs que foram removidas do projeto atual por causa de 'Aceitar item alterado', também serão indicadas pela cor amarela no projeto de referência.

Nota: a aceitação de diferentes partes do projeto (diferenças) ou propriedades de direito de acesso só serão possíveis a partir do projeto de referência do projeto atual. O contrário não vale.

'Extras' 'Aceitar propriedades'

Este comando está disponível no modo comparação (consultar 'Projeto' 'Comparar').

As propriedades do objeto para a POU, onde o cursor está posicionado no momento, serão aceitas para o projeto atual como foram configuradas na versão de referência (a aceitação é possível somente nesta direção!).

'Extras' 'Aceitar direitos de acesso'

Este comando está disponível no modo comparação (consultar 'Projeto' 'Comparar').

Os direitos de acessos para a POU, onde o cursor está posicionado no momento, serão aceitos para o projeto atual como foram configurados na versão de referência.

Menu ‘Projeto’ ‘Mesclar’

Com este comando pode-se mesclar objetos (POUs, tipos de dados, visualizações e recursos) e links com bibliotecas de outros projetos no projeto.

Quando o comando tiver sido selecionado, primeiramente aparecerá a caixa de diálogo padrão para abrir arquivos. Após escolhido o arquivo, uma caixa de diálogo será exibida e nela deve ser selecionado o objeto desejado. A seleção ocorre conforme descrito em 'Projeto' 'Documentar'.

Se um objeto com o mesmo nome já existe no projeto, então, o nome do novo objeto recebe a adição de um sublinhado e um dígito ("_1", "_2"...).

Menu ‘Projeto’ ‘Informações do projeto’

Sob este item do menu, pode-se salvar as informações sobre seu projeto. Quando o comando é selecionado, a caixa de diálogo mostrada na figura abaixo aparece.

As seguintes informações do projeto são exibidas:

- **Nome** do arquivo;
- Caminho do diretório;
- A hora da alteração mais recente (**Data de alteração**).

Essas informações não podem ser alteradas.

Caixa de diálogo para inserir informações do projeto:

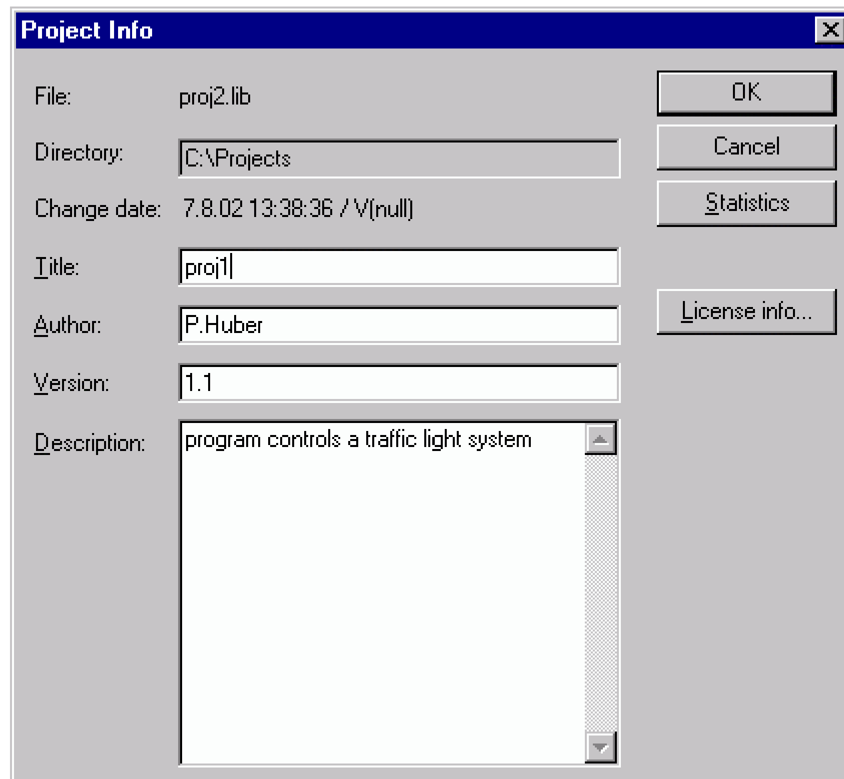


Figura 4-40. Caixa de diálogo para inserir informações do projeto

Além disso, pode-se adicionar as seguintes informações:

Um título do projeto: se suportado pelo sistema do dispositivo, o título proposto será, automaticamente, o nome do arquivo do projeto, no momento em que esse (projeto) é carregado pelo comando 'Arquivo' 'Abrir projeto a partir do CP' (nesse caso a caixa de diálogo para salvar um arquivo será aberta).

O nome do **autor.**

O número da **versão**.

Uma **descrição** do projeto.

Essas informações são opcionais. Ao pressionar o botão '**Estatísticas**', são exibidas informações estatísticas sobre o projeto (número das POUs, tipos de dados e variáveis locais/ globais determinadas na última compilação).

Exemplo de estatísticas do projeto:

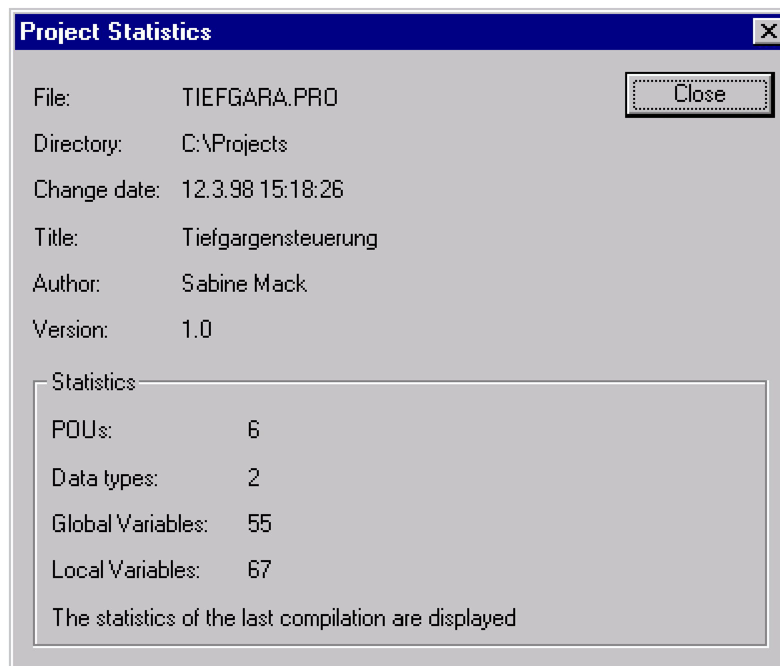


Figura 4-41. Exemplo de estatísticas do projeto

O botão de Informações de licença estará disponível ao trabalhar em um projeto MasterTool IEC, que tenha sido salvo já com informações de licenciamento pelo comando 'Arquivo' 'Salvar como...'.

Ao escolher a opção 'Informações do projeto' na categoria Carregar & Salvar na caixa de diálogo 'Opções', então, ao salvar um novo projeto, ou ao salvar um projeto sob um novo nome, a caixa de diálogo de informações do projeto é chamada automaticamente.

Menu 'Projeto' 'Procura global'

Com este comando pode-se pesquisar a localização de um texto nas POUs, tipo de dados ou nos objetos das variáveis globais.

Ao selecionar este comando, será aberta uma caixa de diálogo, onde pode-se escolher o objeto desejado. A seleção é feita conforme descrito em 'Projeto' 'Documentar'.

Se a seleção é confirmada com **OK**, a caixa de diálogo padrão para a pesquisa será aberta. Ela aparecerá imediatamente quando o comando 'Pesquisa Global' é executado via símbolo na barra de menu. A pesquisa é então realizada em todas as partes do projeto que podem ser pesquisadas. As strings de pesquisa mais recentemente inseridas podem ser selecionadas através da caixa de combinação de **pesquisas por** campo. Se uma string de texto for localizada em um objeto, esse é carregado no editor correspondente ou no gerenciador de bibliotecas e o local onde ela foi encontrada será exibido. A exibição do texto encontrado, bem como a pesquisa e a localização das próximas funções comportam-se de forma similar ao comando 'Editar' 'Pesquisar'.

Ao selecionar o botão J**anela de mensagens**, todos os locais onde a série de símbolos pesquisados no objeto selecionado será listada, linha por linha, em forma de tabela na janela de mensagens. Posteriormente, o número de locais encontrados será exibido.

Se a janela de informações não for aberta, ela será exibida. Para cada local encontrado, será exibido:

- Nome do objeto;
- Localização nas partes de declaração (Dcl) ou implementação (Impl) de uma POU;
- Linha e número da rede se houverem;
- A linha completa nos editores de texto;
- Elementos de texto completos nos editores gráficos.

Janela de mensagens com exibição de resultado de pesquisa:

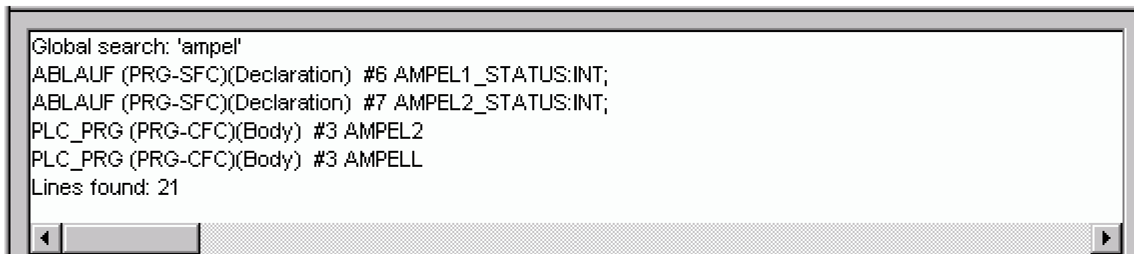


Figura 4-42. Janela de mensagens com exibição de resultado de pesquisa

Ao clicar duas vezes com o mouse em uma linha na janela de mensagens ou pressionar <Enter>, o editor abre com o objeto carregado. A linha referida no objeto é marcada. Pode-se alternar rapidamente entre as linhas exibidas usando as teclas de função <F4> e <Shift>+<F4>.

Menu 'Projeto' 'Substituição global'

Com este comando pode-se procurar o local de um texto em POUs, tipos de dados ou objetos das variáveis globais e substituí-lo por outro. Isto é executado da mesma forma que ocorre com 'Projeto' 'Pesquisa Global' ou 'Editar' 'Substituir'.

Os resultados são exibidos na janela de mensagens.

As bibliotecas, no entanto, não estão disponíveis para seleção e, portanto, não é possível nenhuma exibição na janela de mensagens.

Menu 'Projeto' 'Verificar'

Este comando fornece funções para verificar a correção semântica do projeto. O status da compilação mais recente será considerado. Se o projeto foi alterado nesse meio tempo, faça uma nova compilação para obter uma verificação atualizada. Aparecerá um aviso apropriado na janela de mensagens.

Um sub-menu listando os comandos a seguir será aberto:

- Variáveis não usadas;
- Sobreposição de áreas de memória;
- Acesso concorrente;
- Múltiplo acesso de escrita na saída.

Os resultados serão exibidos na janela de mensagens.

Considerar: nas opções do projeto, categoria 'Compilar', pode-se definir que essas correções semânticas sejam feitas a cada compilação do projeto automaticamente.

Variáveis não usadas

Esta função no menu 'Projeto' 'Verificar' procura por variáveis que foram declaradas, mas não foram usadas no programa. Elas são externadas pelo nome e linha da POU, por exemplo: PLC_PRG (4) – var1. As variáveis nas bibliotecas não são examinadas.

Os resultados são exibidos na janela de mensagens.

Sobreposição de áreas de memória

Esta função do menu 'Projeto' 'Verificar' testa se, na alocação de variáveis via declaração "AT", surgiram sobreposições em áreas específicas de memória. Por exemplo, uma sobreposição ocorre quando alocamos as variáveis "var1 AT %QB21: INT" e "var2 AT %QD5: DWORD" porque ambas usam o byte 21. A saída aparece assim:

%QB21 é referenciado pelas seguintes variáveis:

PLC_PRG (3): VAR1 AT %QB21

PLC_PRG (7): VAR2 AT %QD5

Os resultados são exibidos na janela de mensagens.

Múltiplos acessos de escrita na saída

Esta função do menu 'Projeto' 'Verificar' procura por áreas de memória para as quais um único projeto obtém acesso de escrita em mais de um local. O resultado aparece como indicado a seguir:

%QB24 é escrita nos seguintes locais:

PLC_PRG (3): %QB24

PLC_PRG.POU1 (8): %QB24

Os resultados são exibidos na janela de mensagens.

Acesso concorrente

Esta função do menu 'Projeto' 'Verificar' procura por áreas de memória de endereços IEC que são referenciados em mais de uma tarefa. Nenhuma distinção é feita aqui entre acesso de leitura e escrita. A saída é, por exemplo...

%MB28 é referenciado nas seguintes tarefas:

TASK1 – PLC_PRG (6): %MB28 [ACESSO SOMENTE LEITURA]

TASK2 – POU1.ACTION (1) %MB28 [ACESSO DE ESCRITA]

Os resultados são exibidos na janela de mensagens.

Grupo de usuários

No MasterTool IEC podem ser configurados até oito grupos de usuários com diferentes direitos de acesso para POUs, tipos de dados, visualizações e recursos. Direitos de acesso para objetos únicos ou para todos eles podem ser estabelecidos. Somente um membro de um determinado grupo de usuários pode abrir um projeto. Um membro de um grupo de usuários deve identificar-se por meio de uma senha.

Os grupos de usuários são numerados de 0 a 7, na qual o grupo 0 tem os direitos de administrador, isso é, somente os membros do grupo 0 podem determinar as senhas e direitos de acesso para todos os grupos e/ou objetos.

Quando um novo projeto é inicializado, então, todas as senhas estão inicialmente vazias. Até que uma senha tenha sido definida para o grupo 0, o projeto é inserido automaticamente como um membro do grupo 0.

Se já existir uma senha para o grupo 0 enquanto o projeto estiver sendo carregado, será solicitada uma senha para todos os grupos na abertura do projeto. Para isso a seguinte caixa de diálogo é exibida:

Caixa de diálogo para entrada de senha:

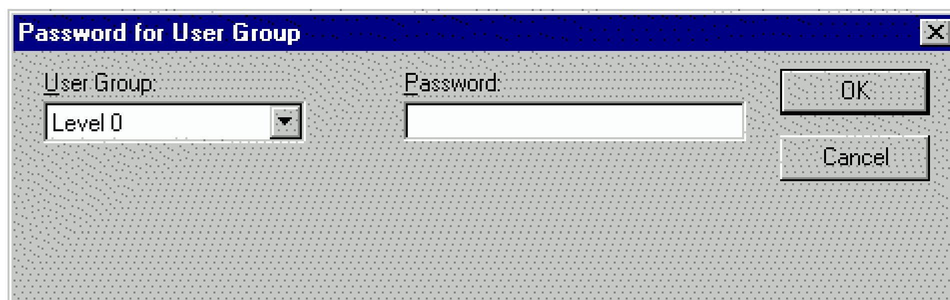


Figura 4-43. Caixa de diálogo para entrada de senha

Na caixa de combinação **Grupo de usuários**, no lado esquerdo da caixa de diálogo, insira o grupo apropriado e digite no lado direito a **senha**. Pressione **OK**. Se a senha não coincidir com a senha salva, então a seguinte mensagem será exibida:

"A senha não está correta."

Somente quando tiver sido inserida a senha correta, o projeto pode ser aberto.

Com o comando 'Senhas para grupo de usuário' são determinadas as senhas, e com 'Objeto' 'Direitos de acesso' são definidos os direitos para objetos únicos ou para todos os elementos.

Menu 'Projeto' 'Senhas de grupo de usuários'

Com este comando obtém-se a caixa de diálogo para atribuição a senha aos grupos de usuários. Este comando só pode ser executado por membros do grupo 0. Quando o comando é executado, a seguinte caixa de diálogo será exibida.

Caixa de diálogo para atribuição de senha:

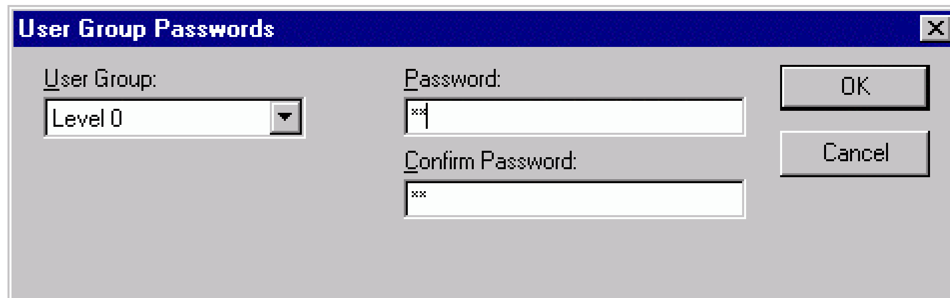


Figura 4-44. Caixa de diálogo para atribuição de senha

Na caixa de combinação **Grupo de usuários**, à esquerda, pode-se selecionar o grupo. Insira a senha desejada para o grupo no campo Senha. Para cada caractere digitado, um asterisco aparecerá (*) no campo. Deve-se repetir a mesma senha no campo **Confirmar senha**. Feche a caixa de diálogo após cada entrada de senha com **OK**. Se aparecer a mensagem:

"A senha não coincide com a confirmação", então ocorreu um erro de digitação durante uma das duas entradas. Nesse caso, repita as duas entradas até que a caixa de diálogo seja fechada sem uma mensagem.

Então, se necessário, atribua uma senha para o próximo grupo chamando o comando novamente.

Importante: se as senhas não forem atribuídas a todos os grupos de usuários, será possível abrir um projeto por meio de um grupo ao qual nenhuma senha foi ainda atribuída!

Use o comando **'Objeto' 'Direitos de acesso'** para atribuir direitos para objetos únicos ou todos eles.

Referente à proteção de um projeto, considerar também o seguinte:

- Senhas para acesso e proteção de gravação (consultar nesse mesmo capítulo: Opções para Senhas);
- Encriptação de um projeto no salvamento (consultar nesse mesmo capítulo: 'Arquivo' 'Salvar Como').

Gerenciando objetos em um projeto

Os próximos capítulos contêm informações sobre como trabalhar com objetos e qual ajuda está disponível para o rastreamento de um projeto (Pastas, Árvore de chamadas, Lista de referência cruzada...).

Objeto

POUs, tipos de dados, visualizações e os recursos variáveis globais, configuração de variáveis, rastreamento de amostragem, configuração do CP, configuração de tarefas, e gerenciador de monitoração e receitas são todos definidos como "objetos". As pastas inseridas para estruturar o projeto estão parcialmente envolvidas. Todos os objetos de um projeto estão no Organizador de Objetos.

Mantendo o ponteiro do mouse por um curto período de tempo em uma POU no Organizador de Objetos, o tipo da POU (Programa, Função ou Bloco funcional) será mostrado em uma tooltip. Para as variáveis globais, a tooltip mostrará a palavra-chave (VAR_GLOBAL, VAR_CONFIG).

Com arrastar e soltar, pode-se deslocar objetos (e também pastas, consultar 'Pasta') dentro de um tipo de objeto. Para isso, selecione o objeto e desloque-o para o ponto desejado, mantendo pressionado o botão esquerdo do mouse. Se o deslocamento resultou em uma colisão de nome, o elemento recém-introduzido será identificado exclusivamente por um número de série acrescentado, (por exemplo, "Object_1").

Pasta

Para controlar projetos maiores, deve-se, sistematicamente, agrupar os POUs, tipos de dados, visualizações e variáveis globais em pastas.

Pode-se configurar quantos níveis de pastas se desejar. Se um sinal de mais estiver na frente de um símbolo com a pasta fechada, então, esta pasta contém objetos e/ou pastas adicionais. Com um clique no sinal de mais a pasta é aberta e os objetos subordinados aparecem. Com um clique no sinal de menos (que substituiu o sinal de mais) a pasta pode ser fechada novamente. No menu de contexto encontram-se os comandos 'Expandir nó' e 'Colapsar nó' com as mesmas funções.

Com arrastar e soltar, pode-se mover os objetos, bem como as pastas dentro do seu tipo de objeto. Para isso, selecione o objeto e arraste-o com o botão esquerdo do mouse pressionado até a posição desejada.

Pode-se criar mais pastas com o comando 'Nova pasta'.

Nota: pastas não têm nenhuma influência sobre o programa e servem apenas para estruturar seu projeto.

Exemplo de pastas no Organizador de Objetos:

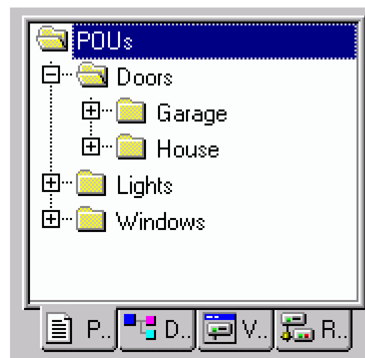


Figura 4-45. Exemplo de pastas no Organizador de objetos

Nova pasta

Com este comando, uma nova pasta é inserida como um objeto estrutural. Se uma pasta tiver sido selecionada, a nova pasta será criada abaixo dela. Caso contrário, ela será criada no mesmo nível. Se uma ação for selecionada, a nova pasta será inserida no nível da POU a qual a ação pertence.

O menu de contexto do Organizador de Objetos que contém este comando será exibido ao selecionar um objeto ou tipo de objeto com o botão direito do mouse ou com <Shift>+<F10>.

A pasta recém-inserida tem, inicialmente, a designação de 'Nova pasta'. Observe a seguinte convenção de nomenclatura para pastas:

Pastas no mesmo nível na hierarquia devem ter nomes distintos. As pastas em níveis diferentes podem ter o mesmo nome.

Uma pasta não pode ter o mesmo nome que um objeto localizado no mesmo nível.

Se já existir uma pasta com o nome da "Nova Pasta" no mesmo nível, a cada pasta adicional com esse nome, automaticamente será acrescentado um número de série (por exemplo, "Nova Pasta 1"). Não é possível renomear para um nome que já esteja sendo usado no momento.

'Expandir nós' 'Colapsar nós'

Com o comando 'Expandir', os objetos selecionados ficam visíveis. Com 'Colapsar', os objetos subordinados ficam ocultos.

As pastas podem ser abertas ou fechadas com um clique duplo do mouse ou pressionando a tecla <Enter>.

O menu de contexto do Organizador de Objetos que contém esse comando é exibido ao selecionar um objeto ou o tipo de objeto e pressionar o botão direito do mouse ou <Shift>+<F10>.

Menu ‘Projeto’ ‘Apagar objeto’

Atalho: <Del>

Com este comando um objeto atualmente selecionado (POU, tipo de dados, visualização ou variáveis globais), ou uma pasta com os objetos subordinados é removido do Organizador de Objetos e, portanto, é excluído do projeto. A exclusão de um objeto pode ser revertida pelo comando 'Editar' 'Desfazer'.

Se a janela do editor do objeto estava aberta, então, ela é automaticamente fechada.

Ao excluir com o comando 'Editar' 'Recortar' , o objeto fica posicionado na área de transferência.

Menu ‘Projeto’ ‘Acrescentar objeto’

Atalho: <Insert>

Com este comando cria-se um novo objeto. O tipo de objeto (POU, tipo de dados, visualização ou variáveis globais) depende do cartão de registro selecionado no Organizado de Objetos. Ao fazer isso, possivelmente será usado um modelo para objetos do tipo 'Variáveis globais', 'Tipos de dados ', 'Função', 'Bloco funcional' ou 'Programa'.

Digite o **Nome da nova POU** na caixa de diálogo que aparece. Lembre-se de que o nome do objeto não deve ter sido usado anteriormente.

Observe as seguintes restrições:

- O nome de uma POU não pode conter espaços;

- Para evitar problemas com as alterações de visualização, uma POU não pode ter o mesmo nome de outra POU, de um tipo de dados nem de uma tela;

- Um tipo de dado não pode receber o mesmo nome de outro tipo de dado ou POU;

- Uma lista de variáveis globais não pode ter o mesmo de outra lista variáveis globais;

- Uma ação não pode ter o mesmo nome de outra ação na mesma POU;

- Para evitar problemas com as alterações de visualização, uma visualização não pode ter o mesmo nome de outra visualização e não deve ter o mesmo nome de uma POU.

Em todos os outros casos, nomes idênticos são permitidos. Assim, por exemplo, ações pertencentes a diferentes POUs podem ter o mesmo nome e uma visualização pode ter o mesmo nome que uma POU.

No caso de uma POU, o seu tipo (programa, função ou bloco funcional) e a linguagem na qual ela é programada também deverão estar selecionados. 'Programa' é o valor padrão do **tipo de POU**, enquanto que a **linguagem da POU** é aquela da POU que foi criada mais recentemente. Se uma POU do tipo 'função' for criada, o tipo de dados desejado deve ser inserido no campo de entrada de texto **Tipo de retorno**. Nesse campo são permitidos todos os tipos de dados, os elementares e os definidos (matrizes, estruturas, enumerações, aliases). O assistente de entrada (<F2>) pode ser usado.

Diálogo para criar uma nova POU:

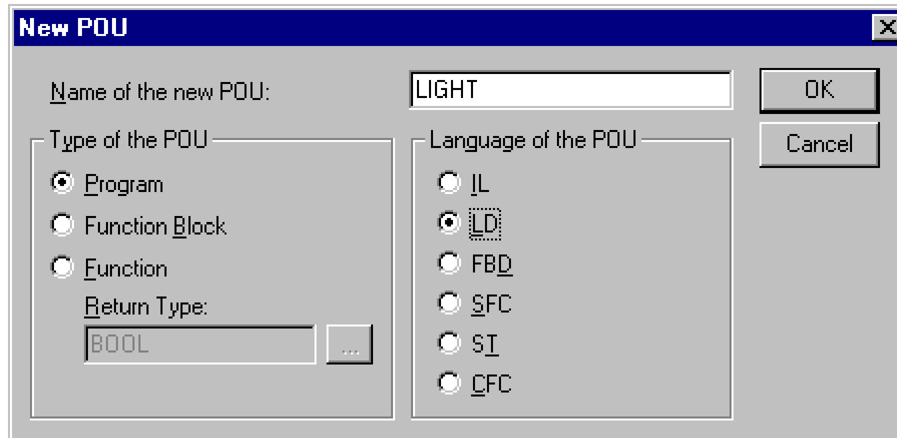


Figura 4-46. Diálogo para criar uma nova POU

Depois de pressionar **OK**, o que só é possível se não houver nenhum conflito de nomenclatura conforme descrito acima, o novo objeto é definido no Organizador de Objetos e a janela de entrada apropriada aparece.

Se o comando **'Editar' 'Inserir'** for usado, o objeto que está na área de transferência será inserido e nenhuma caixa de diálogo aparecerá. Se o nome do objeto inserido apresentar conflito de nomenclatura (veja acima), o comando será executado exclusivamente com adição de um número de série, anexado com um caractere de sublinhado à esquerda (por exemplo, "Rightturnsig_1").

Se o projeto estiver sob o controle de fonte em uma base de dados ENI (dependendo das configurações na caixa de diálogo de opções do projeto para 'Controle de fonte do projeto'), é possível que seja perguntado em qual categoria de base de dados se deseja lidar com o novo objeto. Nesse caso, a caixa de diálogo 'Propriedades' será aberta, onde pode-se atribuir uma das categorias de objeto de base de dados ao objeto.

Salvar como modelo

Objetos do tipo 'Variáveis globais',' Tipos de dados', 'Função', 'Bloco funcional' ou 'Programa' podem ser salvos como modelos. Selecione o objeto no Organizador de Objetos e escolha o comando 'Salvar como modelo' no **menu de contexto** (botão direito do mouse). Dessa forma, cada novo objeto do mesmo tipo irá inicialmente obter a parte da declaração do modelo, automaticamente. Será usado o último modelo criado para um tipo de objeto.

Menu ‘Projeto’ ‘Renomear objeto’

Atalho: <Barra de espaços>

Com este comando se dá um novo nome para o objeto ou pasta atualmente selecionados. Lembre-se de que o nome do objeto não pode ter sido usado.

Se a janela de edição do objeto está aberta, seu título será alterado automaticamente quando o nome for alterado.

Caixa de diálogo para renomear POU:

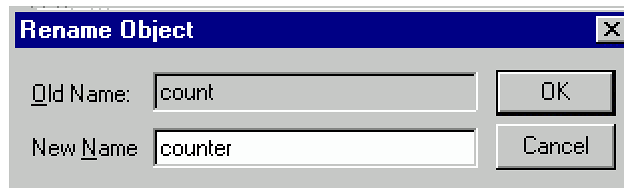


Figura 4-47. Caixa de diálogo para renomear POU

Menu 'Projeto' 'Converter objeto'

Este comando só pode ser usado com POUs. Pode-se converter POUs a partir das linguagens SFC, ST, FBD, LD e IL em uma das três linguagens IL, FBD e LD.

Para tal, o projeto deve estar compilado. Escolha a linguagem para a qual deseja-se converter e dê à POU um novo nome. Lembre-se de que o nome da POU não pode ter sido usado. Em seguida, pressione **OK**, e a nova POU é adicionada à sua lista POU.

O tipo de processamento que ocorre durante a conversão corresponde aquele aplicado na compilação.

Atenção: ações não podem ser convertidas.

Caixa de diálogo para converter uma POU:

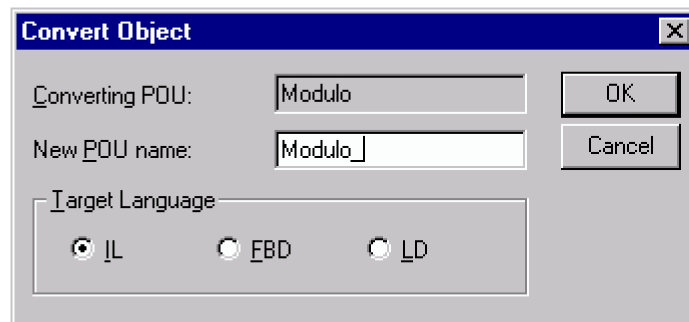


Figura 4-48. Caixa de diálogo para converter uma POU

Considere a seguinte possibilidade: uma POU que foi criada no editor FBD, pode – através do comando 'Extras' 'Exibir' – ser exibida e editada também no Editor LD sem qualquer conversão.

Menu 'Projeto' 'Copiar objeto'

Com este comando um objeto selecionado é copiado e salvo sob um novo nome. Digite o nome do novo objeto na caixa de diálogo que será exibida. Lembre-se de que o nome do objeto não pode ter sido usado.

Se, por outro lado, foi usado o comando **'Editar' 'Copiar'**, o objeto é posicionado na área de transferência sem que nenhuma caixa de diálogo apareça.

Caixa de diálogo para copiar uma POU:

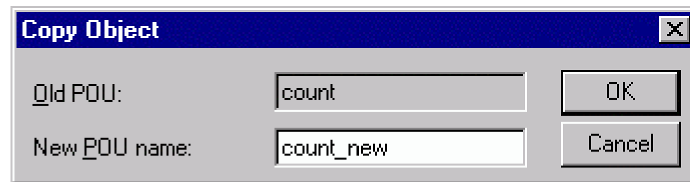


Figura 4-49. Caixa de diálogo para copiar uma POU

Menu ‘Projeto’ ‘Abrir objeto’

Atalho: <Enter>

Com este comando pode-se carregar um objeto selecionado dentro do Organizador de Objetos para o respectivo editor. Se uma janela com esse objeto já estiver aberta, então, ele será focalizado e movido para o primeiro plano, onde então poderá ser editado.

Há duas maneiras de abrir um objeto:

Execute um clique duplo com o mouse no objeto desejado. No Organizador de Objetos, digite a primeira letra do nome do objeto. Isso abrirá uma caixa de diálogo na qual todos os objetos dos tipos de objeto disponíveis que possuem esta letra inicial serão mostrados. Ações são listadas com a notação <nome da POU>.<nome da ação>. Devido ao fato de que os objetos na caixa de diálogo de seleção do objeto estão listados em ordem alfabética, as ações de uma POU sempre são posicionadas abaixo dessa.

Selecione o objeto desejado e clique no botão '**Abrir**' para carregar o objeto na sua janela de edição. Assim, o objeto também fica marcado no Organizador de Objetos e todas as pastas que estão hierarquicamente colocadas acima do objeto são expandidas. Essa opção é suportada com o tipo de objeto "Recursos" e apenas para variáveis globais.

Caixa de diálogo para escolher um objeto a ser aberto:

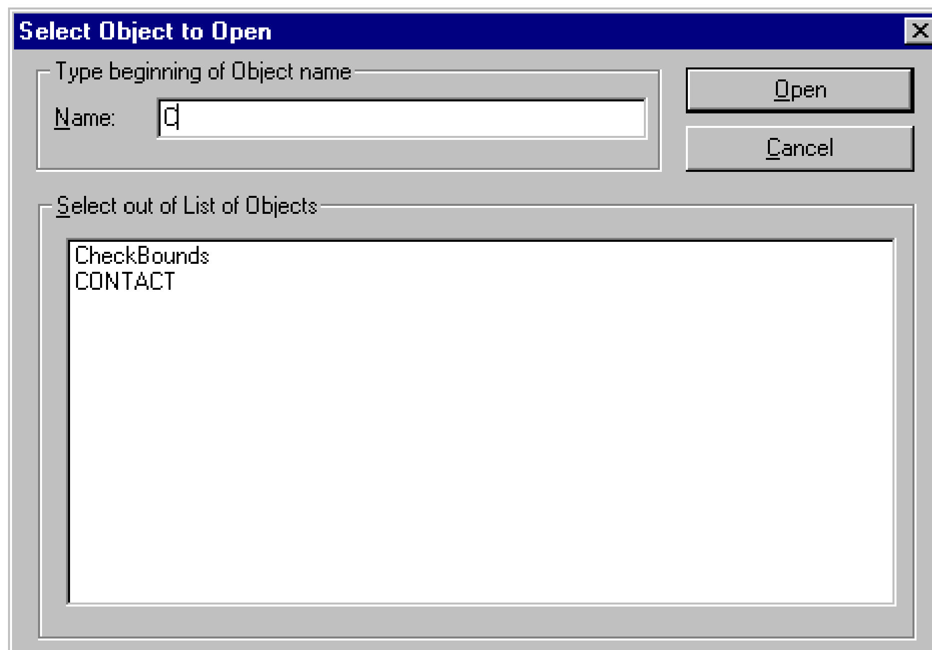


Figura 4-50. Caixa de diálogo para escolher um objeto a ser aberto

Menu ‘Projeto’ ‘Propriedades do objeto’

Este comando abrirá a caixa de diálogo 'Propriedades' para o objeto atual marcado no Organizador de Objetos.

Na guia '**Direitos de Acesso**', encontra-se a mesma caixa de diálogo que aparece ao executar o comando 'Projeto' 'Direitos de acesso a objetos'.

Depende do objeto e das configurações do projeto, se haverão guias adicionais disponíveis para definição das propriedades do objeto...

Lista de variáveis globais

Na guia 'Lista de variáveis globais' os parâmetros relacionados à atualização da lista e troca de dados de variáveis de rede são exibidos e podem ser modificados. Essa caixa de diálogo também será aberta ao criar uma nova lista de variáveis globais, selecionando no Organizador de Objetos uma das entradas na seção 'Variáveis globais ' e executar o comando 'Acrescentar objeto'.

Visualização

Na guia 'Visualização' pode-se definir como deve ser usado o objeto de visualização:

Como usar: se nas configurações do dispositivo, a opção 'Visualização Web' ou 'Visualização do Dispositivo' estiver ativada, pode-se escolher de qual tipo de Visualização o objeto deve ser parte. Usado como: ative uma das seguintes configurações que se referem à possibilidade de usar "Layouts Mestres":

- **Visualização**: o objeto é usado como uma visualização normal;
- **Visualização sem layout mestre**: se um leiaute mestre estiver definido no projeto, ele não será aplicado a este objeto de visualização;
- **Layout mestre**: o objeto será utilizado como leiaute padrão. Por padrão, o layout mestre sempre estará em primeiro plano na visualização, exceto se a opção **plano de fundo** estiver ativada.

Diálogo de Propriedades, Visualização:

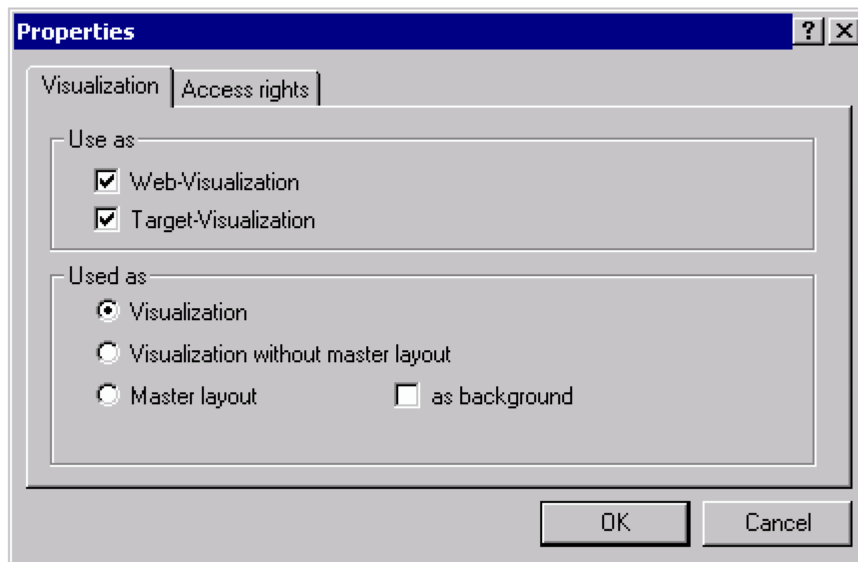


Figura 4-51. Diálogo de Propriedades, Visualização

Menu 'Projeto' 'Direitos de acesso ao objeto'

Com este comando obtém-se a caixa de diálogo que atribui direitos de acesso para diferentes grupos de usuários.

Caixa de diálogo para atribuir direitos de acesso:

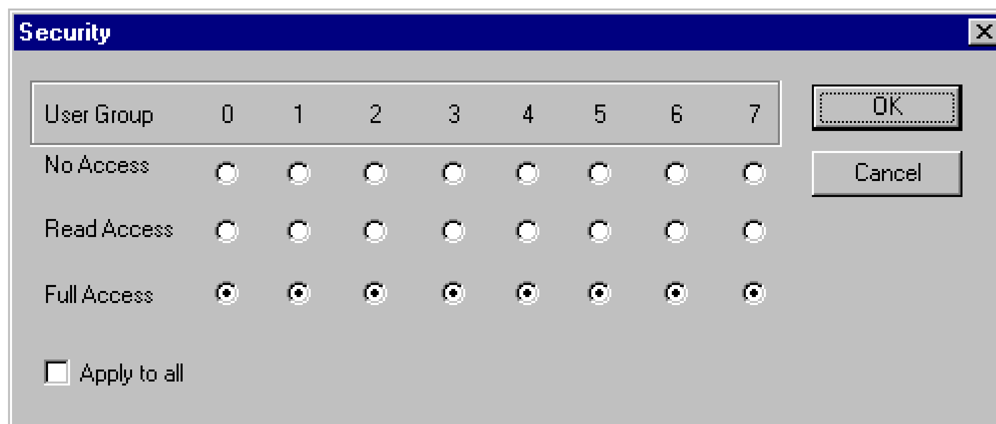


Figura 4-52. Caixa de diálogo para atribuir direitos de acesso

Membros do grupo de usuários 0 agora podem atribuir direitos de acesso individuais para cada grupo de usuários. Há três configurações possíveis:

- **Sem acesso**: o objeto não pode ser aberto por um membro do grupo de usuários;
- **Acesso de leitura**: o objeto pode ser aberto somente para leitura por um membro do grupo de usuários, mas não pode ser alterado;
- **Acesso completo**: o objeto pode ser aberto e alterado por um membro do grupo de usuários.

As configurações referem-se ao objeto atual selecionado no Organizador de Objetos ou, se a opção **Aplicar a todos** for escolhida, para todas as POUs, tipos de dados, visualizações e recursos do projeto também.

A atribuição para um grupo de usuários ocorre quando abre-se o projeto por meio de uma solicitação de senha, se uma senha foi atribuída ao grupo de usuários 0.

Considerar também a possibilidade de atribuir direitos de acesso relacionados à operação de elementos de visualização (Visualização, Segurança).

Menu 'Projeto' 'Acrescentar ação'

Este comando é usado para gerar uma ação em um bloco determinado no Organizador de Objetos. Seleciona-se o nome da ação na caixa de diálogo que aparece e também a linguagem na qual a ação deve ser implementada.

A nova ação é colocada sob o seu bloco no Organizador de Objetos. Um sinal de adição aparece na frente do bloco. Um simples clique do mouse no sinal de mais fará com que os objetos de ação apareçam e, então, o sinal será imediatamente substituído por um sinal de menos. Clicar novamente no sinal de menos faz com que as ações desapareçam e o sinal de adição reapareça. Isso também pode ser feito através dos comandos do menu de contexto 'Expandir nós' e 'Colapsar nós'.

Menu 'Projeto' 'Visualizar instância'

Com este comando é possível abrir e exibir a instância do bloco funcional selecionado no Organizador de Objetos. Da mesma maneira, um clique duplo no bloco funcional no Organizador de Objetos fornece acesso a uma caixa de diálogo de seleção na qual são listadas as instâncias do bloco funcional e a implementação. Selecione aqui a instância ou a implementação desejada e confirme usando OK. O item desejado, então, é exibido em uma janela.

Considerar: para exibir as instâncias, primeiro é necessário fazer login (o projeto deve ter sido compilado sem erros e enviado para o CP com 'Comunicação' 'Login').

Caixa de diálogo para exibir instância:

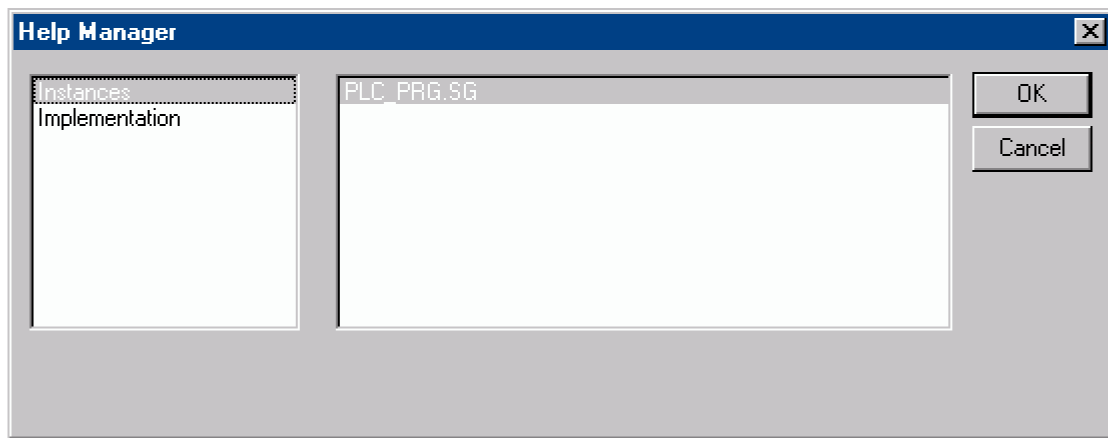


Figura 4-53. Caixa de diálogo para exibir instância

Menu 'Projeto' 'Mostrar árvore de chamadas'

Com este comando obtém-se uma janela que mostra a árvore de chamadas do objeto escolhido no Organizador de Objetos. Antes disso, o projeto deve ter sido compilado sem qualquer erro (consultar 'Compilar tudo'). A árvore de chamadas contém ambas as chamadas para POUs e referências a tipos de dados.

Exemplo de uma árvore de chamadas:

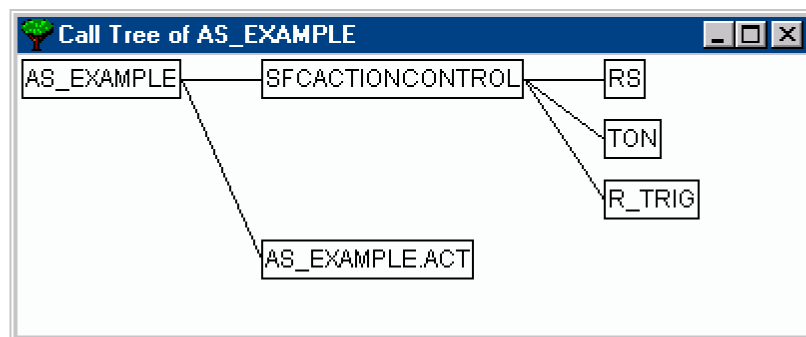


Figura 4-54. Exemplo de uma árvore de chamadas

Menu 'Projeto' 'Mostrar referência cruzada'

Com este comando obtém-se uma caixa de diálogo que torna possível a saída de todos os itens (pontos) da aplicação para uma variável, endereço ou POU. Para isso, o projeto deve ser compilado (consultar 'Projeto' 'Compilar').

Escolha a **categoria** 'Variável', 'Endereço' ou 'POU' primeiro e, então, insira o **nome** do elemento desejado (o assistente de entrada <F2> pode ser usado). Para obter todos os elementos da categoria inserida digite um "*" no nome.

Se o projeto foi alterado desde a última compilação, o termo "(Não atualizado)" será exibido na barra de título da caixa de diálogo. Nesse caso quaisquer referências cruzadas que tenham sido criadas recentemente não serão consideradas na lista, a menos que se proceda uma nova compilação!

Clicando no botão **Referências cruzada** obtém-se a lista de todos os pontos da aplicação. O nome da variável e a informação de endereço, se houver, são especificados juntamente com a POU e a linha ou número de rede. O Espaço de Domínio mostra se essa é uma variável local ou global; a coluna Access mostra se a variável deve ser acessada para 'leitura' ou 'escrita' no local atual. A largura da coluna será adaptada automaticamente para o comprimento das entradas.

Ao selecionar uma linha da lista de referência cruzada e pressionar o botão **Ir Para** ou executar um duplo clique na linha, a POU é mostrada no seu editor no ponto correspondente. Dessa forma pode-se ir para todos os pontos da aplicação sem uma pesquisa demorada.

Visando facilitar o processamento, pode-se usar o botão "**Enviar para a janela de mensagens**" para colocar a lista de referência cruzada atual na janela de mensagens e, a partir deste local, alterar para a respectiva POU.

Caixa de diálogo e exemplo de uma lista de referência cruzada:

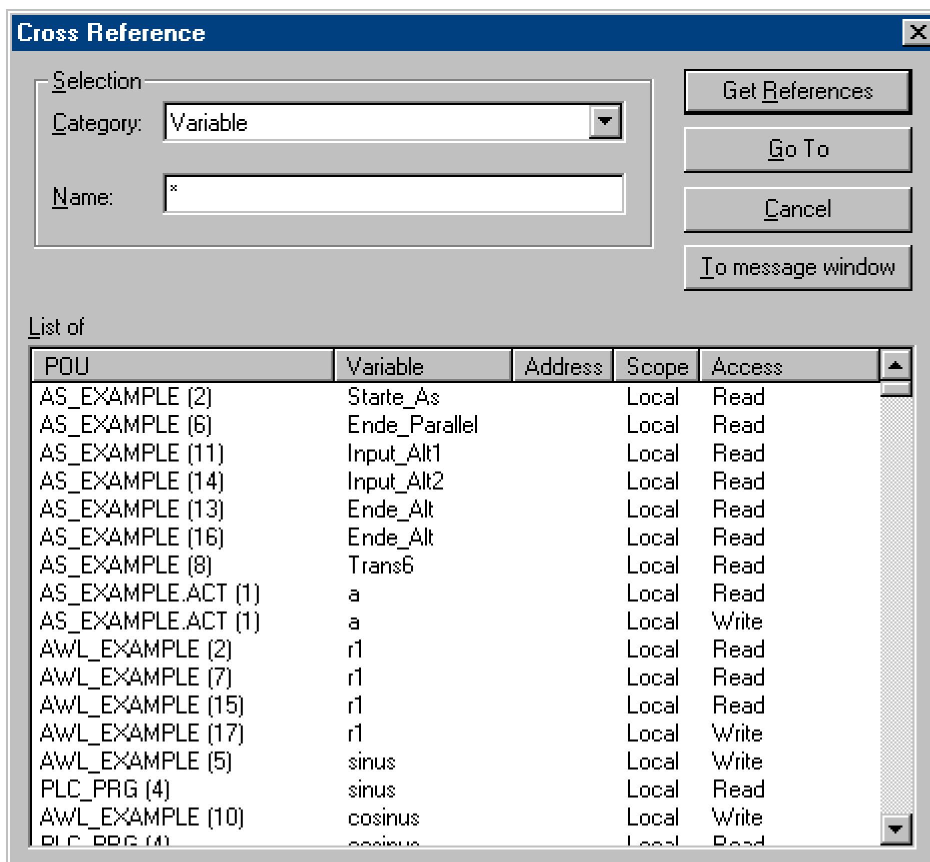


Figura 4-55. Caixa de diálogo e exemplo de uma lista de referência cruzada

Funções de edição gerais

Pode-se usar os comandos a seguir em todos os editores e alguns deles no Organizador de Objetos. Os comandos são localizados sob o item do menu **'Editar'** e no menu de contexto que é aberto com o botão direito do mouse.

Se o software IntelliPoint estiver instalado no computador, o MasterTool IEC oferecerá suporte a todas as funções do Microsoft IntelliMouse.

Em todos os editores que contêm a funcionalidade zoom é possível a ampliação e a redução. Para ampliar pressione a tecla <Ctrl> enquanto movimenta o mouse (scroll); para reduzir movimente retroativamente mantendo a tecla <Ctrl> pressionada.

Menu ‘Editar’ ‘Desfazer’

Atalho: <Ctrl>+<Z>

Este comando desfaz a última ação que foi executada na janela do editor atualmente aberto ou no Organizador de Objetos; o uso repetido desfaz todas as ações de volta para a hora que a janela foi aberta. Isso se aplica a todas as ações em editores para POUs, tipos de dados, visualizações e variáveis globais e no Organizador de objetos.

Com 'Editar' 'Refazer' pode-se restaurar uma ação que tenha sido desfeita.

Nota: os comandos **Desfazer** e **Refazer** se aplicam à janela atual. Cada janela executa sua própria lista de ação. Para desfazer ações em várias janelas, deve-se ativar a janela correspondente. Ao desfazer ou refazer no Organizador de Objetos, o foco deve estar lá.

Menu 'Editar' 'Refazer'

Atalho: <Ctrl>+<Y>

Com este comando na janela do editor atualmente aberto ou no Organizador de objetos, pode-se restaurar uma ação que tenha sido desfeita ('Editar' 'Desfazer').

Sempre que, anteriormente, tiver sido executado o comando **'Desfazer'**, o comando **'Refazer'** estará disponível.

Nota: os comandos **'Desfazer'** e **'Refazer'** aplicam-se à janela atual. Cada janela executa sua própria lista de ação. Para desfazer ações em várias janelas, então, deve-se ativar a janela correspondente. Ao desfazer ou refazer no Organizador de objetos, o foco deve estar lá.

Menu 'Editar' 'Recortar'

Símbolo:

Atalho: <Ctrl>+<X> ou <Shift>+<Delete>

Este comando transfere a seleção atual do editor para a área de transferência. A seleção será removida a partir do editor.

Com o Organizador de Objetos isso se aplica da mesma forma que com objeto selecionado, porém nem todos os objetos podem ser recortados, como por exemplo, a Configuração do CP.

Lembre-se de que nem todos os editores oferecem suporte a este comando "Recortar" e que o mesmo pode ser limitado em alguns editores.

A forma da seleção depende do respectivo editor:

Nos editores de texto IL, ST e declarações a seleção é uma lista de caracteres.

Nos editores FBD e LD a seleção é um número de redes indicadas por um retângulo pontilhado no campo de número de rede ou por uma caixa com todas as linhas anteriores, caixas e operandos.

No editor SFC a seleção é uma parte de uma série de passos circundados por um retângulo pontilhado.

Para colar o conteúdo da área de transferência use o comando 'Editar' 'Colar'. No Editor SFC também pode-se usar os comandos 'Extras' 'Inserir ramificação paralela (direita)' ou 'Extras' 'Colar depois'.

Para copiar uma seleção para a área de transferência sem excluí-la, use o comando 'Editar' 'Copiar'.

Para remover uma área selecionada sem alterar a área de transferência, use o comando 'Editar' 'Excluir'.

Menu 'Editar' 'Copiar'

Símbolo:

Atalho: <Ctrl>+<C>

Este comando copia a seleção atual do editor para a área de transferência. Isso não altera o conteúdo da janela do editor.

Com o Organizador de Objetos isso se aplica da mesma forma que com o objeto selecionado, porém nem todos os objetos podem ser copiados, como por exemplo, a Configuração do CP.

Lembre-se de que nem todos os editores oferecem suporte ao comando "Copiar" e que o mesmo pode ser limitado em alguns editores.

Para o tipo de seleção as mesmas regras se aplicam como ocorre com **'Editar' 'Recortar'**.

A forma de seleção depende do respectivo editor:

Nos editores de texto (IL, ST e declarações) a seleção é uma lista de caracteres.

Nos editores FBD e LD a seleção é um número de redes indicadas por um retângulo pontilhado no campo de número de rede ou por uma caixa com todas as linhas anteriores, caixas e operandos.

No editor SFC a seleção é uma parte de uma série de passos circundados por um retângulo pontilhado.

Para colar o conteúdo da área de transferência use o comando 'Editar' 'Colar'. No Editor SFC também pode-se usar os comandos 'Extras' 'Inserir ramificação paralela (direita)' ou 'Extras' 'Colar depois'.

Para excluir uma área selecionada e simultaneamente colocá-la na área de transferência, use o comando 'Editar' 'Recortar'.

Menu 'Editar' 'Colar'

Símbolo:

Atalho: <Ctrl>+<V>

Este comando cola o conteúdo da área de transferência na posição atual da janela do editor. Nos editores gráficos o comando só pode ser executado quando uma estrutura correta resulta da inserção.

Com o Organizador de Objetos, o objeto é colado a partir da área de transferência.

Lembre-se de que nem todos os editores oferecem suporte ao comando "Colar" e que o mesmo pode ser limitado em alguns editores.

A posição atual pode ser definida diferentemente de acordo com o tipo de editor:

Nos editores de texto (IL, ST e declarações) a posição atual é aquela do cursor intermitente (uma linha vertical) que pode ser obtida clicando com o mouse.

Nos editores FBD e LD a posição atual é a primeira rede com um retângulo pontilhado na área do número da rede. O conteúdo da área de transferência é inserido na frente dessa rede. Se for copiada uma estrutura parcial, ela será inserida na frente do elemento selecionado.

No Editor SFC a posição atual é determinada pela seleção circundada por um retângulo pontilhado. Dependendo da seleção e do conteúdo da área de transferência, os conteúdos são inseridos na frente da seleção ou em uma nova ramificação (paralela ou alternativa) à esquerda da seleção.

No SFC, os comandos 'Extras' 'Inserir ramificação paralela (direita)'ou 'Extras 'Colar depois' podem ser usados para inserir o conteúdo da área de transferência.

Para copiar uma seleção para a área de transferência sem excluí-la, use o comando 'Editar' 'Copiar'.

Para remover uma área selecionada sem alterar a área de transferência, use o comando 'Editar' 'Excluir'.

Menu 'Editar' 'Excluir'

Atalho: <Del>

O comando exclui a área selecionada da janela do editor. Isso não altera o conteúdo da área de transferência.

Com o Organizador de Objetos isso se aplica da mesma forma que com o objeto selecionado, porém nem todos os objetos podem ser excluídos, como por exemplo, a Configuração do CP.

Para o tipo de seleção se aplicam as mesmas regras que em **'Editar' 'Recortar'**.

A forma de seleção depende do respectivo editor:

Nos editores de texto (IL, ST e declarações) a seleção é uma lista de caracteres.

Nos editores FBD e LD a escolha é um número de redes, as quais são indicadas por um retângulo pontilhado no campo de número de rede ou por uma caixa com todas as linhas anteriores, caixas e operandos.

No editor SFC a seleção é uma parte de uma série de passos circundados por um retângulo pontilhado.

No gerenciador de bibliotecas a seleção é o nome da biblioteca selecionada no momento.

Para excluir uma área selecionada e simultaneamente colocá-la na área de transferência, use o comando 'Editar' 'Recortar'.

Menu 'Editar' 'Localizar'

Símbolo:

Com este comando pode-se procurar por uma determinada passagem de texto na janela do editor atual. A caixa de diálogo 'Localizar' é aberta e assim permanece até que o botão **Cancelar** seja pressionado.

No campo **Localizar** pode-se digitar a série de caracteres que está procurando.

Além disso, pode-se decidir se o texto que se está procurando deve **coincidir com a palavra inteira** ou não, se letras **maiúsculas de minúsculas** devem ser diferenciadas e também se a pesquisa deve continuar **para cima** ou **para baixo** a partir de posição atual do cursor.

O botão **Localizar Próximo** inicia a pesquisa, a qual começa na posição selecionada e continua na direção escolhida. Se as passagens de texto foram encontradas, elas são realçadas. Se a passagem não for encontrada, então aparecerá uma mensagem. A pesquisa pode ser repetida várias vezes sucessivamente até que o início ou fim do conteúdo da janela do Editor tenha sido encontrado. No Editor CFC a ordem geométrica dos elementos será considerada e a pesquisa será executada do canto superior esquerdo da janela para o canto superior direito. Considerar que POUs FBD são processadas da direita para a esquerda!

Caixa de diálogo Localizar:

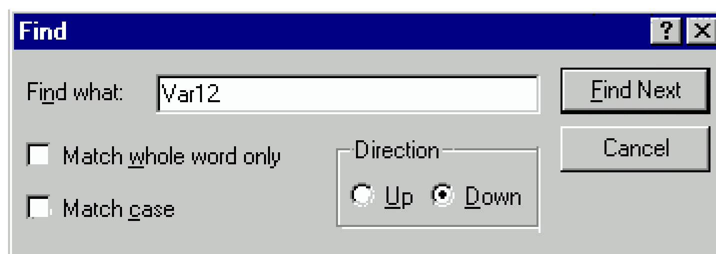


Figura 4-56. Caixa de diálogo Localizar

Menu 'Editar' 'Localizar próximo'

Símbolo:

Atalho: <F3>

Com este comando pode-se executar uma pesquisa com os mesmos parâmetros da última ação realizada em 'Editar' 'Localizar'. Considerar que POUs FBD são processadas da direita para a esquerda!

Menu 'Editar' 'Substituir'

Com este comando pode-se procurar por uma determinada passagem, da mesma forma que com o comando 'Editar' 'Localizar', e a substitui por outra. Após a seleção deste comando uma caixa de diálogo para localizar e substituir é exibida. Ela permanece aberta até que o botão **Cancelar** ou **Fechar** seja pressionado.

Caixa de diálogo Localizar e Substituir:

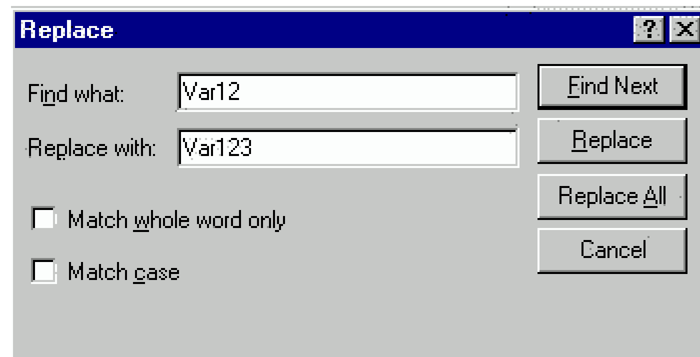


Figura 4-57. Caixa de diálogo Localizar e Substituir

No campo anterior a "**Localizar**", a seqüência de caracteres que estiver marcada no editor antes será inserida automaticamente. Também pode-se inserir a seqüência de pesquisa manualmente. Pressionando o botão **Substituir**, ocorrerá a substituição da seleção atual pelos caracteres fornecidos no campo **Substituir por**. Use o botão **Localizar Próximo** para ir para a próxima passagem onde a seqüência de caracteres for encontrada. Considerar que POUs FBD são processadas da direita para a esquerda!

O botão **Substituir Todos** substitui todas as ocorrências do texto no campo **Localizar Próximo** depois da posição atual pelo texto do campo **Substituir Por**. Ao final do procedimento uma mensagem informa quantas substituições foram feitas.

Menu 'Editar' 'Assistente de entrada'

Atalho: <F2>

Este comando oferece uma caixa de diálogo para escolher entradas possíveis na posição atual do cursor na janela do editor. Na coluna esquerda, escolha a categoria de entrada desejada, selecione essa entrada na coluna à direita e confirme sua escolha com **OK**. Isto insere sua opção nessa posição.

As categorias oferecidas dependem da posição atual do cursor na janela do editor, isto é, o que pode ser inserido nesse ponto (por exemplo, variáveis, operadores, POUs, conversões, etc..).

Se a opção **Com Argumentos** estiver ativa, então quando o elemento selecionado for inserido, os argumentos a serem transferidos são especificados com (por exemplo): bloco funcional fu1 selecionado, o qual define a variável de entrada var_in: fu1(var_in:=);

Inserção de função Func1, que usa var1 e var2 como parâmetros de transferência: Func1(var1,var2)

É possível alternar a exibição dos elementos disponíveis em estruturada e não estruturada. Isso ocorre por meio de ativação/desativação da opção **Exibição estruturada**.

Nota: para inserir identificadores também pode ser usada a funcionalidade IntelliSense.

Exibição não estruturada

Caixa de diálogo para assistência de entrada não estruturada:

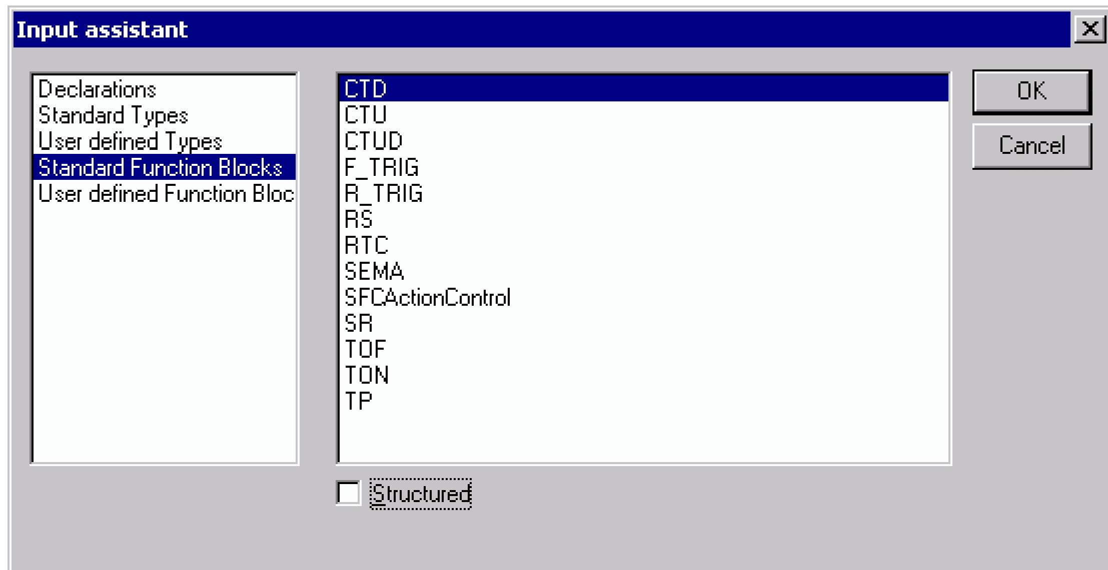


Figura 4-58. Caixa de diálogo para assistência de entrada não estruturada

As POUs, variáveis ou tipos de dados em cada categoria são simplesmente classificados linearmente em ordem alfabética.

Em vários locais (por exemplo, na lista de monitoração), nomes de variáveis multi-estágio são necessárias. Nesses eventos, a caixa de diálogo 'Assistente de Entrada' exibe uma lista de todas as POUs e também um ponto único para as variáveis globais. Após cada nome da POU existe um ponto. Se uma POU é selecionada por um duplo clique ou pressionando <Enter>, a lista de variáveis que pertencem a ela será aberta. Se estiverem presentes instâncias e tipos de dados, será possível abrir mais níveis na hierarquia de exibição. **OK** transfere a variável selecionada mais recentemente.

Pode-se alternar para exibição estruturada por meio de ativação de **Exibição Estruturada**.

Exibição estruturada

Se a Exibição **Estruturada** estiver marcada, as POUs, variáveis ou tipos de dados serão classificados hierarquicamente. Esta opção é possível para programas padrão, funções padrão, blocos funcionais padrão, programas definidos, funções definidas, blocos funcionais definidos, variáveis globais, variáveis locais, tipos definidos e variáveis de monitoração. A exibição visual e hierárquica corresponde à exibição do Organizador de Objetos; se forem referenciados elementos em uma biblioteca, esses serão inseridos em ordem alfabética na parte superior e a hierarquia pertinente será exibida como no Gerenciador de Bibliotecas.

As **variáveis de entrada e saída** de blocos funcionais que são declaradas como variáveis locais ou globais são listadas na categoria 'Variáveis locais' ou 'Variáveis globais' sob o nome da instância (ex.: Inst_TP ET, Inst_TP in,...). Para chegar lá, selecione o nome da instância (ex.: Inst_TP) e confirme com OK.

Se a **instância de um bloco funcional** está selecionada, a opção **Com Argumentos** pode ser usada. Nas linguagens de texto ST e IL bem como durante a configuração da tarefa, o nome da instância e os parâmetros de entrada do bloco funcional, são, então, inseridos.

Por exemplo, se Inst (Dec. inst.: TON;) é selecionado, o que segue é inserido:

```
INST (IN:= ,PT:=)
```

Se a opção não estiver selecionada, somente o nome da instância será inserido. Nas linguagens gráficas ou na janela de monitoração de variáveis, geralmente, somente o nome da instância é inserido.

Caixa de diálogo para assistência de entrada estruturada:

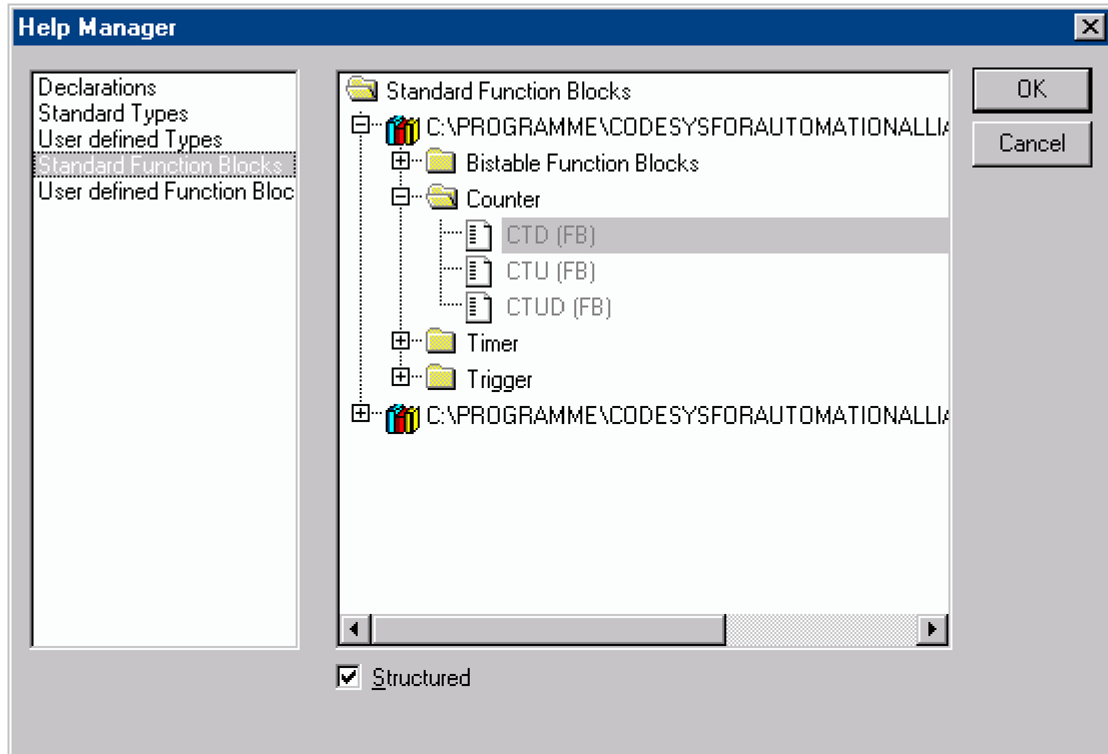


Figura 4-59. Caixa de diálogo para assistência de entrada estruturada

Componentes de **estruturas** são exibidos de forma análoga às instâncias de bloco funcional.

Para **enumerações**, os valores de enumeração individuais são listados sob o tipo de enumeração. A ordem é: enumerações de bibliotecas, enumerações de tipos de dados, enumerações locais das POUs.

A regra geral é que linhas contendo sub-objetos não são selecionáveis (exceto instâncias, veja acima), porém podem ter sua hierarquia de exibição expandida ou reduzida por um nível, como para os nomes de variáveis multi-estágios.

Se o assistente de entrada for chamado no 'Gerenciador de Monitoração e Receitas' ou na seleção de variáveis de rastreamento na caixa de diálogo de configuração do rastreamento, é possível fazer uma **seleção múltipla**. Quando a tecla <Shift> é pressionada, pode-se selecionar um intervalo de variáveis; quando a tecla <Ctrl> é pressionada pode-se selecionar várias variáveis individuais. As variáveis selecionadas são, então, marcadas. Se, durante a seleção de intervalo, estiverem selecionadas linhas que não contêm variáveis válidas (por exemplo, nomes de POU), essas linhas não serão incluídas na seleção. Quando as seleções individuais são feitas, essas linhas não podem ser marcadas.

Com a caixa de diálogo do Assistente de entrada, é possível transferir estruturas, matrizes ou instâncias na **janela de monitoração** e na **configuração de rastreamento**. Como um clique duplo com o botão do mouse é associado à extensão ou redução da exibição de hierarquia do elemento, a seleção nesses casos só pode ser confirmada com **OK**.

Daí em diante, as variáveis selecionadas são inseridas linha por linha na janela de **monitoração**, isto é, cada variável selecionada é gravada em uma linha separada. No caso das variáveis de rastreamento, cada variável é inserida em uma linha separada da lista.

Se o número máximo de variáveis de rastreamento (20) for excedido durante a inserção das variáveis selecionadas, a mensagem de erro “O número máximo de variáveis permitidas é 20" aparece. Outras variáveis selecionadas não serão, então, inseridas na lista.

Pode-se alternar para exibição não estruturada por meio da desativação da opção **Exibição Estruturada**.

Nota: algumas entradas (por exemplo, variáveis globais) são atualizadas na caixa de diálogo do Assistente de Entrada somente após a compilação.

Menu 'Editar' 'Auto-declarar'

Atalho: <Shift>+<F2>

Este comando abre a caixa de diálogo para a declaração de uma variável. Essa caixa de diálogo também se abre automaticamente quando a opção 'Projeto' 'Opções 'Editar' 'Auto-declarar' for selecionada e quando uma nova variável indefinida é usada no editor de declaração.

Menu 'Editar' 'Próximo erro'

Atalho: <F4>

Após a compilação incorreta de um projeto, este comando mostra o próximo erro. A janela do editor correspondente é ativada e o local incorreto é selecionado. Ao mesmo tempo, na janela de mensagens, a mensagem de erro correspondente é exibida.

Menu 'Editar' 'Erro anterior'

Atalho: <Shift>+<F4>

Após a compilação incorreta de um projeto este comando exibe o erro anterior. A janela do editor correspondente é ativada e o local incorreto é selecionado. Ao mesmo tempo, na janela de mensagens, a mensagem de erro correspondente é mostrada.

Menu 'Editar' 'Macros'

Este item do menu leva a uma lista de todas as macros definidas para o projeto (para informações sobre a geração de macros, consultar 'Projeto' 'Opções' 'Macros'). Quando uma macro executável é selecionada, se abre uma caixa de diálogo 'Processar macro'. O nome da macro e a linha de comando ativa no momento são exibidos. O botão Cancelar pode ser usado para interromper o processamento da macro. Nesse evento o processamento do comando atual será concluído mesmo assim. Em seguida será exibida uma mensagem apropriada na janela de mensagens e no LOG durante a operação online: "<Macro>: execução interrompida pelo usuário".

As macros podem ser executadas offline e online, mas em cada caso, apenas são executados os comandos que estão disponíveis no respectivo modo.

Funções online gerais

Os comandos online disponíveis estão no item do menu **'Comunicação'**. A execução de alguns dos comandos depende do editor ativo. Consultar os capítulos seguintes para uma descrição.

Os comandos online ficam disponíveis somente depois do login. Consultar 'Funções online gerais...' na guia Conteúdo desta ajuda online.

Graças à funcionalidade 'Alteração online' existe a possibilidade de fazer alterações nos programas no controlador em execução. Consultar 'Comunicação' 'Login'.

Consultar mais adiante, o diagrama que mostra as relações entre o Projeto-Compilar, Projeto-Download, Alteração Online e Login.

Menu 'Comunicação' 'Login'

Símbolo:

Atalho: <Alt>+<F8>

Este comando combina o sistema de programação com o CP (ou inicia o programa de simulação) e altera-o para o modo online.

Se o projeto atual não foi compilado desde a abertura ou desde a última modificação, ele é compilado agora (como em 'Projeto' 'Compilar'). Se ocorrerem erros durante a compilação, o MasterTool IEC não passará para o modo online.

Se o projeto atual foi alterado no controlador desde o último download, mas não foi fechado e, se as últimas informações de download não foram excluídas com o comando 'Projeto' 'Limpar tudo', então, após o comando 'Login' uma caixa de diálogo se abre com a pergunta: "O programa foi alterado. Carregar alterações? (**Alteração online**)". Ao responder "**Sim**" confirma-se que, no login, as partes modificadas do projeto serão carregadas para o controlador (consultar abaixo o diagrama de relações entre Projeto-Compilação, Projeto-Download, Alterações Online e Login e também algumas dicas em Alterações Online). A resposta "**Não**" resulta no login sem as alterações feitas desde o último download. A opção **Cancelar**, cancela o comando. <Carregar tudo> faz com que o projeto inteiro seja recarregado no controlador.

Caixa de diálogo estendida de Login:

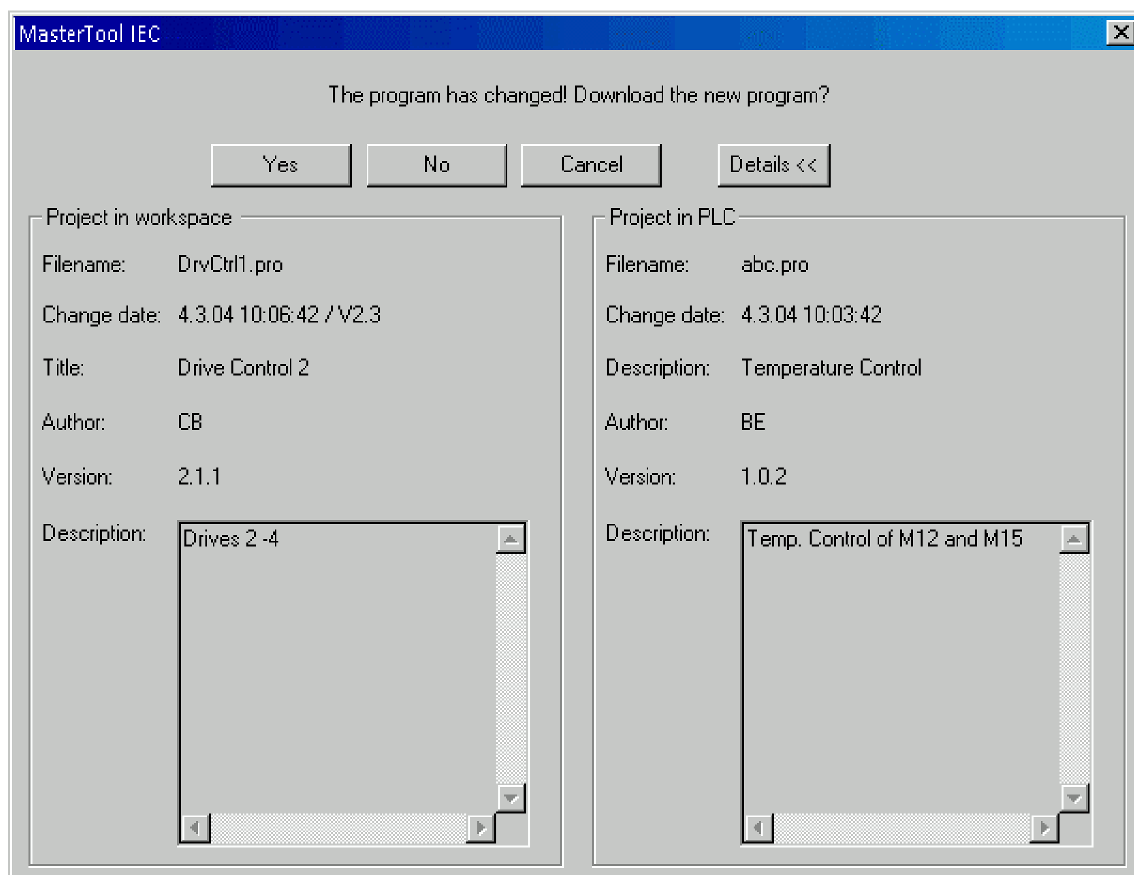


Figura 4-60. Caixa de diálogo estendida de Login

Se nas opções do projeto, categoria Ambiente de trabalho, a opção 'Comunicação no modo de segurança' estiver ativada e, se o sistema do dispositivo suportar a funcionalidade, no diálogo do Login – automaticamente – as informações do projeto também serão exibidas. Estas serão as informações do projeto atualmente aberto no MasterTool IEC e que já está disponível no controlador. Por meio do botão **Detalhes** << pode-se fechar esta parte das informações da caixa de diálogo.

Se a opção 'Comunicação no modo de segurança' não estiver ativada, pode-se abrir explicitamente a exibição de informações do projeto na caixa de diálogo, através do botão **Detalhes** >>**.**

Por favor, observe as seguintes considerações:

- depende do dispositivo qual botão está definido como botão padrão

- não é possível realizar alterações online após modificações na Configuração de Tarefas ou do CP, após a inserção de uma biblioteca e nem após a execução de 'Projeto' 'Limpar tudo' (veja abaixo);

- a alteração online não causa uma re-inicialização das variáveis, assim, as modificações dos valores de inicialização não serão consideradas! Variáveis retentivas mantêm seus valores quando uma alteração online é realizada, porém elas não se comportam desta maneira no novo download do projeto (veja abaixo, 'Comunicação' 'Download').

Após um login bem-sucedido, todas as funções online estão disponíveis (se as configurações correspondentes foram inseridas no 'Projeto' 'Opções' categoria 'Compilar'). Os valores atuais são monitorados para todas as declarações de variáveis visíveis.

Use o comando 'Comunicação' 'Logout' para retornar do modo online para o modo offline.

Relações entre Login/ Compilar/ Download/ Alterações Online

Veja abaixo um diagrama que mostra as relações entre Login, Compilar, Download e Alterações Online:

Os seguintes termos são usados:

- Fonte - projeto atual do MasterTool IEC (arquivo *.pro, PC local);
- Compilar - compilar informações a partir do último processo compilado é necessário para compilação incremental (arquivo *.ci, PC local);
- Download- informações sobre o que estava carregado no CP no último download (arquivo*.ri, PC local);
- CP - projeto atualmente disponível no CP arquivo *.prg, sistema do dispositivo).

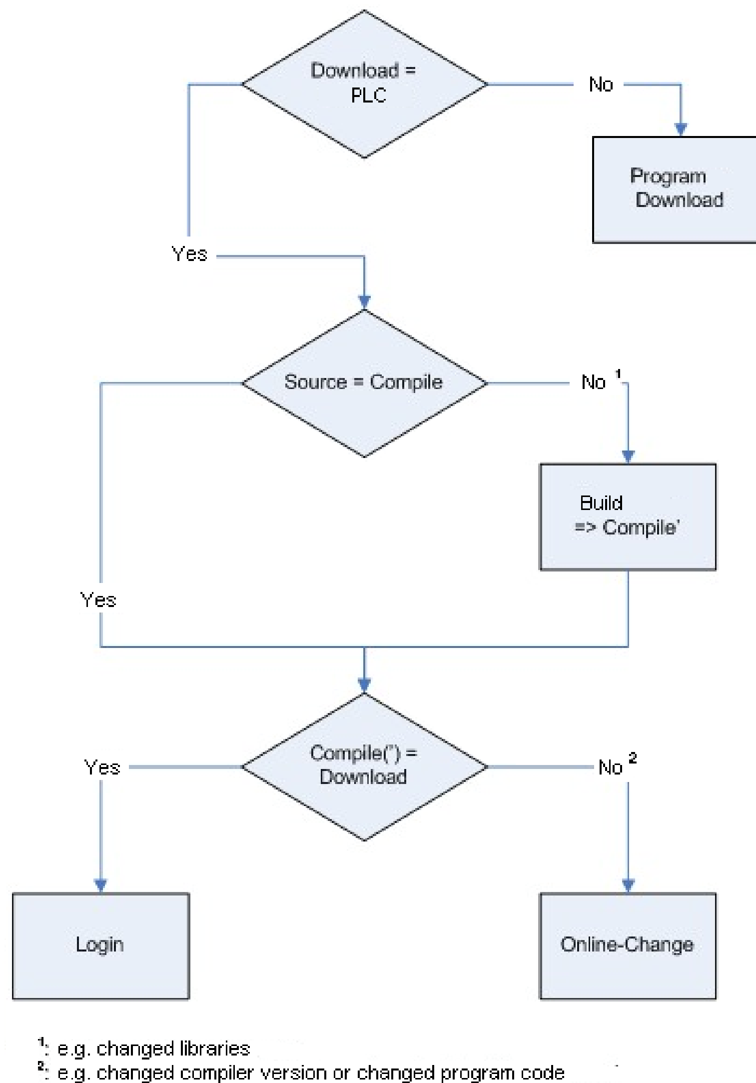


Figura 4-61. Diagrama que mostra as relações entre Login, Compilar, Download e Alterações online

Dicas de alterações online

Não é possível realizar alterações online após modificações na Configuração de Tarefas ou do CP , após a inserção de uma biblioteca e nem após a execução de 'Projeto' 'Limpar tudo' (veja abaixo).

Se as informações do download (arquivo <nome do projeto> <identificador do dispositivo>.ri), criadas no último download do projeto (pode ter sido uma alteração online também), foram excluídas nesse meio tempo (ex.: por meio do comando 'Projeto' 'Limpar tudo'), então, não será mais possível realizar alterações online.

Essa regra, entretanto, apresenta, uma exceção: se o arquivo-ri foi salvo em outro local ou foi renomeado e, portanto, ainda está disponível, ele pode ser carregado explicitamente pelo comando 'Projeto' 'Carregar informações de download'. Em relação a isso consultar também abaixo 'Alterações online em um projeto... '.

Alterações online não causam uma re-inicialização das variáveis. Assim, as modificações dos valores de inicialização não serão consideradas!

Variáveis retentivas mantêm seus valores quando uma alteração online é realizada, porém elas não se comportam desta maneira no novo download do projeto (veja abaixo, 'Comunicação' 'Download').

Alterações online para um projeto que está sendo executado em vários CPs

Para executar um projeto proj.pro em dois controladores idênticos CP1 e CP2 (mesmo sistema do dispositivo) e certificar-se de que pode ser feita uma atualização do projeto em ambos os controladores por meio da alteração online, faça o seguinte:

(1) Carregue e inicialize o projeto no CP1, salvando as informações de download para CP1

1. Conecte o projeto MasterTool IEC proj.pro ao controlador CP1 (Comunicação/Parâmetros de comunicação) e carregue proj.pro no CP1 (Comunicação/Login, Download). No download, o arquivo proj00000001.ri será criado no diretório de projetos, contendo informações de download.

2. Renomeie o proj00000001.ri, por exemplo, para proj00000001_CP1.ri. É necessário salvar o arquivo com outro nome porque em um próximo download de proj.pro, o arquivo proj00000001.ri (em outro CP) será substituído com novas informações de download e, portanto, as informações pertencentes ao download em CP1 seriam perdidas.

3. Inicie o projeto no CP1 e efetue logout ('Comunicação' 'Iniciar', 'Comunicação' 'Logout').

(2) Carregue e inicialize o projeto no CP2, salvando as informações de download para o CP2

1. Agora, conecte-se ao controlador CP2 (usando o mesmo dispositivo que o CP1) e carregue o proj.pro no CP2. Assim, novamente será criado um arquivo proj00000001.ri no diretório de projetos, contendo as informações sobre o download feito no momento.

Renomeie o novo proj00000001.ri, por exemplo para proj00000001_CP2.ri.

3. Inicie o projeto no CP2 e efetue logout ('Comunicação' 'Iniciar', 'Comunicação' 'Logout').

(3) Altere o projeto no MasterTool IEC

1. No MasterTool IEC, agora faça as modificações no proj.pro que se deseja transferir posteriormente, via Alterações online para o programa executado em ambos os CPs.

(4) Alterações Online no CP1, Novo salvamento de informações de download para CP1

1. Para tornar possível a alteração online para proj.pro no CP1, primeiramente as informações de download referentes ao download do proj.pro no CP1 devem ser restauradas. No login,o MasterTool IEC está procurando por um arquivo proj00000001.ri. Porém as informações de download apropriadas estão armazenadas no arquivo proj00000001_CP1.ri.

Agora existem 2 possibilidades:

(a) Pode-se renomear proj00000001_CP1.ri novamente para proj00000001.ri. Assim, em um login no CP1, automaticamente as informações de download apropriadas estão disponíveis e o MasterTool IEC perguntará se deseja-se fazer uma alteração online.

(b) Em vez disto pode-se explicitamente carregar o arquivo proj00000001_CP1.ri antes do login, usando o comando 'Projeto' 'Carregar informações de download'. Nesse caso, não é necessário renomear o arquivo.ri e não será mais possível realizar uma alteração online.

2. Na alteração online no CP1, uma versão atualizada do arquivo proj00000001.ri foi criada, contendo as informações de download. Armazene este arquivo novamente, conforme descrito em (4), para mantê-lo disponível para outra alteração online no CP1.

(5) Alterações online no CP2, Novo salvamento de informações sobre o download para CP2

Para tornar possível uma alteração online, relacionada às modificações no proj.pro feito em (3) também no CP2, por favor, execute as etapas correspondentes para proj00000001_CP2.ri conforme descrito na etapa (4).

(6) Para cada Alteração Online após uma modificação no projeto: execute os passos de (3) a (5)

Reportes do sistema

Erro:

"O perfil do controlador selecionado não corresponde ao sistema do dispositivo..."

Verifique se o sistema do dispositivo inserido nas configurações do sistema do dispositivo (Recursos) corresponde aos parâmetros inseridos em 'Comunicação' 'Parâmetros de comunicação'.

Erro:

"Erro de comunicação. Ocorreu logout."

Verifique se o controlador está em execução. Verifique se os parâmetros inseridos em **'Comunicação' 'Parâmetros de comunicação'** correspondem àqueles do seu controlador. Em particular, deve-se verificar se a porta correta foi inserida e se as velocidades de comunicação no controlador e no sistema de programação conferem. Se o servidor de gateway for usado, verifique se o canal correto está definido.

Erro:

"O programa foi modificado! O novo programa deve ser carregado?"

O projeto que é aberto no editor é incompatível com o programa atual encontrado no CP (ou com o programa que está sendo executado no modo de Simulação). Monitoração e depuração, portanto, não são possíveis. Pode-se escolher "Não", realizar logout e abrir o projeto correto ou escolher a opção "Sim" para carregar o projeto atual no CP.

Mensagem:

"O programa foi alterado. Carregar alterações? (ALTERAÇÃO ONLINE)".

O projeto está em execução no controlador. O sistema do dispositivo suporta as 'Alterações online' e o projeto foi alterado no controlador (alterações do último download ou da última alteração online). Agora pode-se decidir se essas alterações devem ser carregadas com o programa do controlador em execução ou se o comando deve ser cancelado. Pode-se também, no entanto, carregar todo código compilado, selecionando o botão **Carregar todos**.

Menu ‘Comunicação’ ‘Logout’

Símbolo:

Atalho: <Ctrl>+<F8>

A conexão com o CP é interrompida, ou o programa no modo de Simulação é encerrado e retorna para o modo offline.

O comando 'Comunicação' 'Login' permite a passagem para o modo online.

Menu ‘Comunicação’ ‘Download’

Este comando carrega o projeto compilado no CP.

Ao usar geração de código C, então, antes do download, o compilador C é chamado para criar o arquivo de download. Se isso não for o caso, o arquivo de download é criado durante a compilação.

Durante a compilação, as informações sobre download são salvas em um arquivo chamado **<nome do projeto> 0000000ar.ri**, o qual é usado na Alteração online para comparar o programa atual com o último (mais recente) programa carregado para o controlador. Dessa forma, somente os componentes alterados do programa são recarregados. Este arquivo é apagado pelo comando 'Projeto' 'Limpar tudo'. Em relação a alterações online em vários CPs, por favor, consultar: 'Comunicação’ 'Login'. Considere que o arquivo *.ri também é atualizado durante uma alteração online.

Dependendo das configurações do sistema do dispositivo a cada criação de um projeto de inicialização no modo offline, o arquivo *.ri pode ser gerado novamente.

Somente as variáveis persistentes mantêm seus valores, mesmo após um download (veja capítulo 5: Variáveis Remanentes).

Menu ‘Comunicação’ ‘RUN’

Símbolo:

Atalho:<F5>

Este comando inicia o programa no CP ou no modo de Simulação. Ele pode ser executado:

- imediatamente após o comando 'Comunicação' 'Download';

- após o usuário no CP ter finalizado o programa através do comando 'Comunicação' 'STOP';

- quando o usuário do programa está em um breakpoint ou ainda;

- quando o comando 'Comunicação' 'Ciclado' foi executado.

Menu ‘Comunicação’ ‘STOP’

Símbolo:

Atalho: <Shift>+<F8>

Interrompe a execução do programa no CP ou no modo de Simulação entre dois ciclos.

Use o comando 'Comunicação' 'RUN' para continuar o programa.

Menu ‘Comunicação’ ‘Reset’

Exceto para as variáveis retentivas (VAR RETAIN), com este comando pode-se fazer o reset de todas as variáveis para o valor específico, com o qual elas foram inicializadas (inclusive aquelas variáveis que foram declaradas como VARIÁVEIS PERSISTENTES!). Se as variáveis foram inicializadas com um valor específico, este comando redefinirá as variáveis para o seu valor de início. Todas as outras variáveis são definidas em uma inicialização padrão (por exemplo, inteiros em 0). Como uma medida de precaução, o MasterTool IEC pede para confirmar sua decisão antes de todas as variáveis serem substituídas. Essa é uma situação que ocorre no caso de uma falha de energia ou quando se desativa o Controlador e o ativa novamente logo em seguida (reset a quente) enquanto o programa está em execução.

Use o comando 'Comunicação' 'RUN' para reiniciar o programa.

Consultar também 'Comunicação' 'Reset (original)', 'Comunicação' 'Reset (a frio)' e, para obter uma visão geral sobre reinicialização, consultar: Variáveis Remanentes, no capítulo 5.

Menu ‘Comunicação’ ‘Reset (a frio)’

Este comando corresponde ao 'Reset', mas, além das variáveis "normais" e persistentes, também as variáveis retentivas (!) retornam para seus valores de inicialização. Essa situação ocorre no início de um programa que foi enviado pouco antes para CP (início a frio).

Consultar também 'Comunicação' 'Reset', 'Comunicação' 'Reset (original)' e para obter uma visão geral sobre reinicialização, consultar Variáveis Remanentes.

Menu ‘Comunicação’ ‘Reset (original)’

Este comando redefine todas as variáveis, incluindo as remanentes (VAR RETAIN e VAR PERSISTENT) para seus valores de inicialização e apaga o programa do usuário no controlador. O controlador retorna ao seu estado original.

Consultar também: 'Comunicação' 'Reset', 'Comunicação' 'Reset (a frio)' e, para obter uma visão geral sobre reinicialização, consultar: Variáveis Remanentes, no capítulo 5.

Menu ‘Comunicação’ ‘Alternar breakpoint’

Símbolo:

Atalho: <F9>

Este comando define um breakpoint na posição atual na janela ativa. Se um breakpoint já tiver sido definido na posição atual, esse breakpoint será removido.

A posição na qual é possível definir um breakpoint depende da linguagem na qual a POU está escrita na janela ativa.

Nos Editores de Texto (IL, ST), o breakpoint é definido na linha onde o cursor está localizado, se essa linha for uma posição de breakpoint (reconhecida pela cor cinza escuro no campo de número de linha). Também pode-se clicar no campo de número de linha para definir ou remover um breakpoint em editores de texto.

No FBD e LD, o breakpoint é definido na rede atualmente selecionada. Para definir ou remover um breakpoint nesses editores, também pode-se clicar no campo de número da rede.

No SFC, o breakpoint é definido no passo selecionado no momento. No SFC também pode-se usar <Shift> com um clique duplo para definir ou remover um breakpoint.

Se um breakpoint foi definido, o campo do número de linha ou o campo do número de rede ou o passo serão exibidos com uma cor de plano de fundo azul claro.

Se um breakpoint foi definido enquanto o programa está em execução, o programa irá parar, e o campo correspondente será exibido em uma cor de plano de fundo vermelho. Para continuar o programa, usando os comandos 'Comunicação' 'RUN', 'Comunicação' 'Passo a passo' ou 'Comunicação' 'Sobrepasso'.

Também pode-se usar a caixa de diálogo do breakpoint para definir ou remover breakpoints.

Menu ‘Comunicação’ ‘Diálogo de breakpoint’

Este comando abre uma caixa de diálogo para editar os breakpoints no projeto inteiro. A caixa de diálogo também exibe todos os breakpoints atualmente definidos.

Para definir um breakpoint, escolha uma **POU** na caixa de combinação da POU e a linha ou a rede na caixa de combinação (localização) onde se deseja definir o breakpoint; a seguir, pressione o botão **Acrescentar**. O breakpoint será adicionado à lista.

Para excluir um breakpoint, realce o breakpoint a ser excluído da lista e pressione o botão **Excluir**.

O botão **Excluir tudo** pode ser usado para excluir todos os breakpoints.

Para ir para o local no editor onde um determinado breakpoint estava definido, realce o breakpoint respectivo da lista e pressione o botão **Ir para**.

Caixa de diálogo para edição de breakpoint:

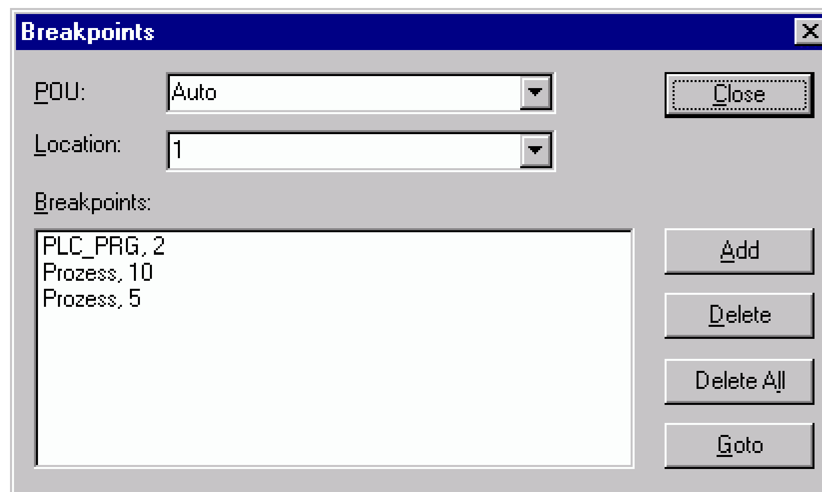


Figura 4-62. Caixa de diálogo para edição de breakpoint

Para definir ou excluir breakpoints, também pode-se usar o comando 'Comunicação' 'Alternar breakpoint'.

Menu 'Comunicação' 'Sobrepasso'

Símbolo:

Atalho: <F10>

Este comando faz com que um passo único seja executado. Se uma POU é chamada, o programa pára após sua execução. No SFC uma ação completa é executada.

Se a instrução atual é o chamado de uma função ou de um bloco funcional, então, a função ou um bloco funcional serão executados completamente. Use o comando 'Comunicação' 'Passo a passo', para mover-se para a primeira instrução de uma função de chamada ou bloco funcional.

Se a última instrução foi atingida, então, o programa irá para a próxima instrução na POU.

Menu 'Comunicação' 'Passo a passo'

Atalho: <F8>

Um passo único é executado. O programa é interrompido antes da primeira instrução de uma POU chamada.

Se necessário, haverá uma mudança para uma POU aberta.

Se a posição atual é um chamado de função ou de bloco funcional, então, o comando prosseguirá para a primeira instrução na POU chamada.

Em todas as outras situações, o comando funcionará exatamente como 'Comunicação' 'Sobrepasso'.

Menu 'Comunicação' 'Ciclado'

Atalho: <Ctrl>+<F5>

Este comando executa um único ciclo do CP e pára após esse ciclo.

Ele pode ser repetido continuamente para prosseguir em ciclos únicos.

O modo ciclado termina quando o comando 'Comunicação' 'RUN' é selecionado.

Menu 'Comunicação' 'Escrever valores'

Atalho: <Ctrl>+<F7>

Com este comando, uma ou mais variáveis são definidas - apenas uma vez! – para os valores definidos pelo usuário no início de um ciclo (consultar 'Comunicação' 'Forçar valor (es)' para obter esta definição de forma permanente).

Os valores de todas as variáveis de elemento único podem ser alterados, desde que também estejam visíveis na monitoração.

Antes que o comando 'Escrever valores' seja executado, o valor da variável deve estar pronto para ser escrito.

Para variáveis não-booleanas deve ser executado um duplo clique do mouse na linha em que variável é declarada (ou a variável deve estar marcada e a tecla <Enter> ser pressionada concomitantemente). A caixa de diálogo 'Escrever variável <x>' então aparece, na qual o valor a ser escrito para a variável pode ser inserido.

Diálogo para escrita de variáveis:

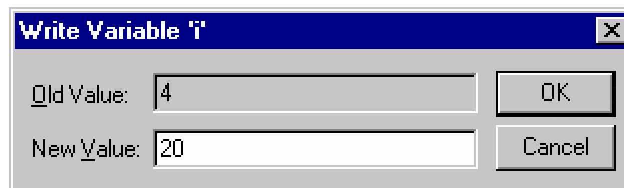


Figura 4-63. Diálogo para escrita de variáveis

Para variáveis booleanas, o valor é alternado (entre TRUE e FALSE, com nenhum outro valor permitido), clicando duas vezes na linha na qual a variável é declarada; nenhuma caixa de diálogo aparece.

O valor definido para a Escrita é exibido na cor turquesa entre parênteses, atrás do primeiro valor da variável. Por exemplo, a = 0 <:=34>.

Dica (exceção): nos Editores FBD e o LD o valor é mostrado na cor turquesa, porém sem parênteses, próximo ao nome da variável.

Defina os valores para as variáveis desejadas.

Os valores inseridos para serem escritos nas variáveis também podem ser corrigidos ou excluídos da mesma maneira. Da mesma forma, isto é possível em 'Comunicação' 'Diálogo escrever/forçar' (veja abaixo).

Os valores a serem escritos são previamente salvos em uma **lista de monitoração**, onde permanecem até que sejam efetivamente gravados, excluídos ou transferidos para uma lista de forçamento pelo comando 'Forçar valor (es)'.

O comando para escrever valores pode ser encontrado em dois lugares:

- Comando 'Escrever valores' no menu 'Comunicação'
- Botão 'Escrever valores' no diálogo 'Editando as listas de escrita/forçamento'

Quando o comando 'Escrever valores' for executado, todos os valores contidos na lista de escrita serão gravados, somente uma vez, nas variáveis apropriadas no controlador, no início do ciclo. Em seguida, serão excluídos da lista de escrita. Se o comando 'Forçar valor (es)' for executado, as variáveis em questão serão também removidas da lista de escrita e transferidas para a lista de forçamento.

Nota: na linguagem SFC, os valores individuais a partir dos quais uma expressão de transição é montada não podem ser alterados com 'Escrever valores'. Isto se deve ao fato de que, na monitoração, é exibido o 'valor total' da expressão, e não os valores das variáveis individuais (ex.: "a AND b" só é exibido como TRUE se ambas as variáveis realmente tiverem o valor TRUE).

Em FBD, por outro lado, somente a primeira variável em uma expressão (usada, por exemplo, como entrada para um bloco funcional) é monitorada. Assim, o comando 'Escrever valores' só é possível para esta variável.

Menu 'Comunicação' 'Forçar valores'

Atalho: <F7>

Com este comando, uma ou mais variáveis são definidas permanentemente para os valores estabelecidos pelo usuário no início de um ciclo (consultar 'Comunicação' 'Escrever valor (es)' para obter esta definição apenas uma vez).

Seqüência de tempo em um ciclo:

1. Ler entradas;
2. Forçar valores;
3. Processar código;
4. Forçar valores;
5. Escrever saídas.

A função permanece ativa até que seja explicitamente suspensa pelo usuário (comando 'Comunicação' 'Liberar forçado(s)') ou ocorra logout do sistema de programação.

Para definir os novos valores primeiramente deve ser criada uma **lista de escrita**, exatamente como descrito em ' Comunicação ' 'Escrever valores'. As variáveis contidas na lista de escrita são marcadas na monitoração. A lista de escrita é transferida para uma **lista de forçamento** assim que o comando ' Comunicação ' 'Forçar valor (es)' é executado. É possível que uma lista de forçamento ativa já exista e, nesse caso, ela é atualizada quando necessário. A lista de escrita é esvaziada e os novos valores são exibidos em vermelho como 'forçados'. Modificações na lista de forçamento são transferidas para o programa no próximo comando 'Forçar valor (es)'.

Nota: a lista de forçamento é criada no primeiro forçamento das variáveis contidas na lista de escrita.

O comando para forçamento de uma variável (inserção na lista de forçamento) pode ser encontrado nos seguintes locais:

- Comando 'Forçar valores' no menu ' Comunicação ';
- Botão 'Forçar valores' na caixa de diálogo 'Editar a lista de escrita e forçamento'.

Nota: na linguagem SFC, os valores individuais a partir dos quais uma expressão de transição é montada, não podem ser alterados com 'Forçar valores'. Isso se deve ao fato de que, na monitoração, o 'valor total' da expressão (e não os valores das variáveis individuais) é exibido (ex.: "a AND b" só é exibida como TRUE se ambas as variáveis realmente tem o valor TRUE).

Em FBD, por outro lado, somente a primeira variável em uma expressão (usada, por exemplo, como entrada para um bloco funcional) é monitorada. Assim, o comando 'Escrever valores' só é possível para esta variável.

Menu 'Comunicação' 'Liberar forçado(s)'

Atalho: <Shift>+<F7>

Este comando finaliza o forçamento de valores das variáveis no controlador. Os valores de variáveis voltam para o valor normal.

Variáveis forçadas podem ser reconhecidas no Monitoramento pela cor vermelha na qual seus valores são exibidos. Pode-se excluir toda a lista de forçamento, mas também pode-se marcar variáveis simples para as quais o forçamento deve ser liberado.

Para excluir a lista de forçamento inteira, o que significa liberar **todas as variáveis forçadas**, escolha uma das seguintes maneiras:

Comando 'Liberar forçado(s)' no menu 'Comunicação'.

Botão 'Liberar forçado(s)' na caixa de diálogo 'Editar a lista de escrita e forçamento'

Excluir a lista de forçamento inteira usando o comando 'Liberar forçado(s)' na caixa de diálogo 'Remover lista de escrita/forçamento'. Essa caixa de diálogo abre ao escolher o comando 'Liberar forçado (s)' enquanto existir também uma lista de escrita.

Para liberar o forçamento apenas **para variáveis únicas** é necessário marcá-las primeiro. Faça isso de uma das maneiras descritas a seguir. Após isto, as variáveis escolhidas são marcadas com uma extensão turquesa <Liberar forçado(s) >:

Um clique duplo do mouse em uma linha, na qual uma variável não booleana é declarada, abre a caixa de diálogo 'Escrever variável <x>'. Pressione o botão <Liberar forçamento para essa variável>.

Repita o duplo clique do mouse em uma linha na qual uma variável booleana é declarada para alternar a exibição <Liberar Forçado(s) > no final da linha.

No menu 'Comunicação' abra a caixa de diálogo 'Escrita/Forçamento' e exclua o valor no campo de edição da coluna 'Valor forçado'.

Quando, para todas as variáveis desejadas, a configuração "<Liberar Forçado(s) >" for mostrada na janela de declaração, escolha o comando 'Forçar valor (es)' para transferir as modificações da lista de forçamento para o programa.

Se a lista de escrita atual (consultar 'Comunicação' 'Escrever valores') não estiver vazia enquanto se executa o comando **'Liberar Forçado(s)'**, a caixa de diálogo 'Remover lista de escrita/forçamento' será aberta. Ali o usuário tem que decidir se ele quer apenas **liberar o forçamento**, remover a lista de escrita ou se, além disso, deseja remover as duas listas.

Caixa de diálogo para remoção de lista de escrita/forçamento:

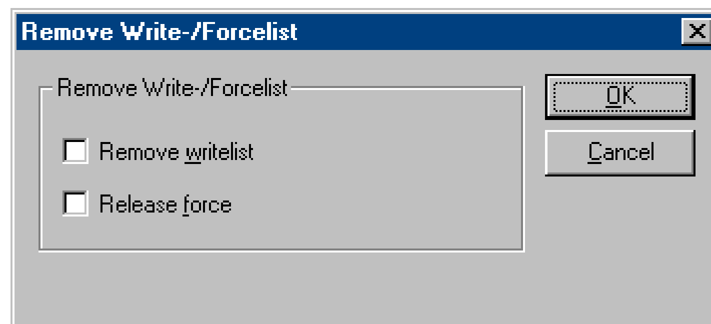


Figura 4-64. Caixa de diálogo para remoção de lista de escrita/forçamento

Menu ‘Comunicação’ ‘Diálogo escrever/forçar’

Atalho: <Ctrl>+<Shift>+<F7>

Este comando leva a uma caixa de diálogo que exibe em dois registros a **lista de escrita** e a **lista de forçamento** atual. Cada nome de variável e o valor a ser escrito ou forçado são exibidos em uma tabela.

Caixa de diálogo para editar a lista de escrita e forçamento:

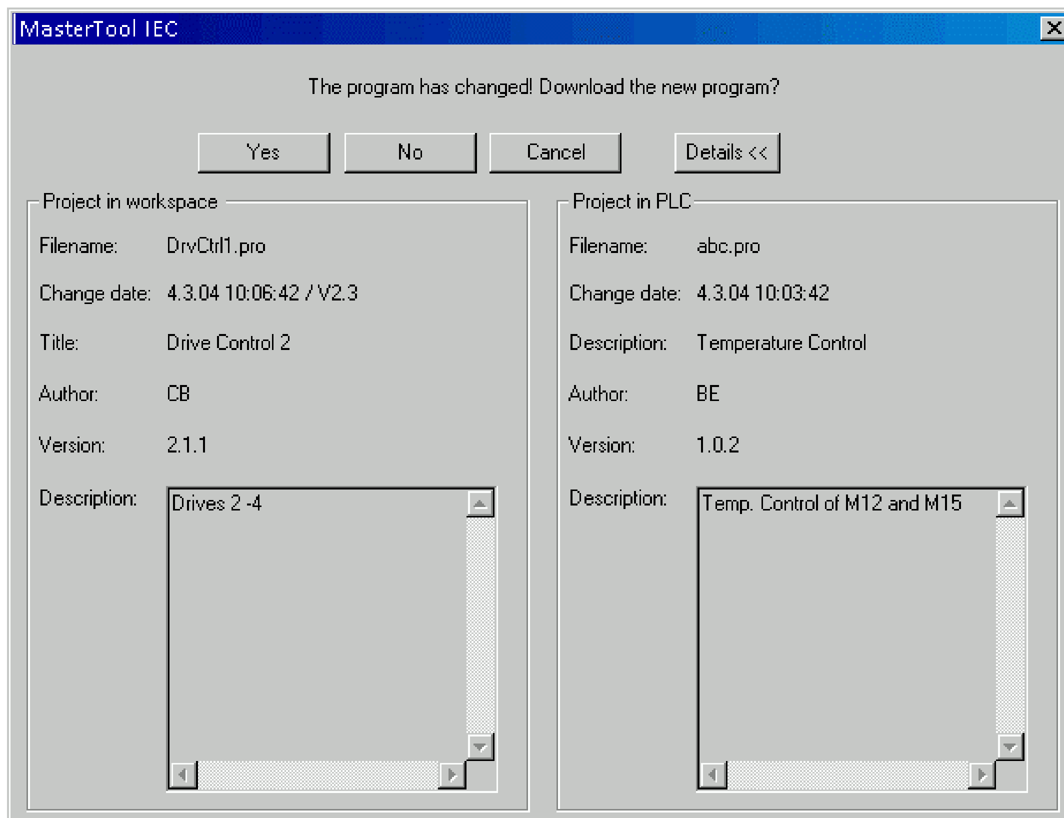


Figura 4-65. Caixa de diálogo para remoção de lista de escrita/forçamento

As variáveis aparecem na lista de escrita através dos comandos 'Comunicação' 'Escrever valor (es)' e são transferidas para a lista de forçamento via comando 'Comunicação' 'Forçar valor (es)'. Os valores podem ser ali editados nas colunas “Valor preparado" ou “Valor forçado”, clicando com o mouse em uma entrada para abrir um campo de edição. Se a entrada não é do tipo consistente, uma mensagem de erro será exibida. Se um valor for excluído, isto significa que a entrada será excluída da lista de escrita ou a variável será indicada para suspensão do forçamento, assim que a caixa de diálogo for fechada com qualquer outro comando, exceto **Cancelar**.

Os comandos seguintes, correspondente àqueles no menu Comunicação, estão disponíveis por meio dos botões:

- **Forçar valor (es)**: todas as entradas na lista de escrita atual são transferidas para a lista de forçamento, isto é, os valores das variáveis no controlador são forçados. Todas as variáveis marcadas com 'Liberar forçado(s)' não são mais forçadas. A caixa de diálogo é então fechada;
- **Escrever valor (es)**: todas as entradas na lista de escrita atual são gravadas somente uma vez para as variáveis correspondentes no controlador. A caixa de diálogo é então fechada;
- **Liberar forçado(s)**: todas as entradas na lista de forçamento serão excluídas ou, se uma lista de escrita é apresentada, a caixa de diálogo "Excluir lista de escrita/forçamento" aparece, na qual o usuário deve decidir se ele quer apenas **liberar os valores forçados**, **descartar a lista de escrita**, ou ambos. A caixa de diálogo fechará nesse ponto ou depois que a caixa de diálogo de seleção for fechada, conforme o caso.

Menu ‘Comunicação’ ‘Mostrar pilha de chamadas’

Pode-se executar este comando quando o modo de Simulação pára em um breakpoint. Aparecerá uma caixa de diálogo com uma lista da pilha de chamadas de POUs.

Exemplo de uma pilha de chamadas:

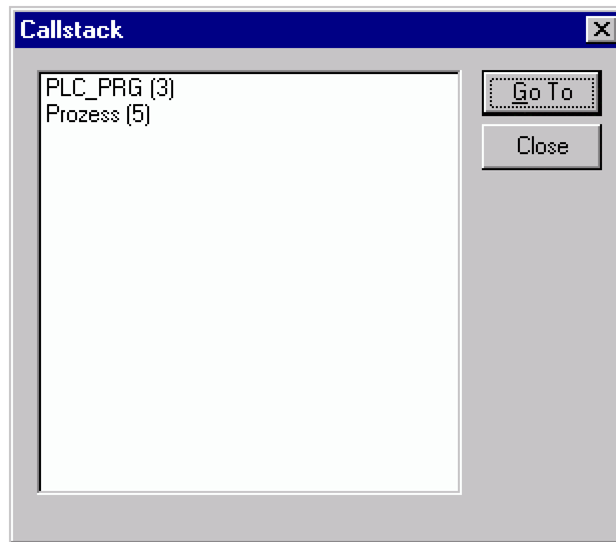


Figura 4-66. Exemplo de uma pilha de chamadas

A primeira POU é sempre PLC_PRG, porque é onde começa a execução.

A última POU é sempre a POU que está sendo executada.

Depois que foi selecionada uma POU e pressionado o botão “Ir para”, ela é carregada em seu editor, e exibirá a linha ou rede sendo processadas.

Menu ‘Comunicação’ ‘Exibir controle de fluxo’

Dependendo das configurações do sistema do dispositivo, o usuário pode ativar e desativar a função de Controle de Fluxo. Se ela está ativada, um sinal de verificação (✓) aparecerá na frente do item do menu. A seguir, todas as linhas ou redes executadas no último ciclo do CP serão marcadas.

O campo do número da linha ou o campo do número da rede das linhas ou redes executadas será exibido em verde. Um campo adicional é adicionado no Editor IL, no qual o conteúdo presente do acumulador é exibido. Nos editores gráficos (FBD e LD), um campo adicional será inserido em todas as linhas de conexão que não transportarem valores booleanos. Quando essas Entradas e Saídas são verificadas, o valor transportado através da linha de conexão será mostrado nesse campo. Conectando-se as linhas que transportam somente valores booleanos, as mesmas estarão em azul sombreado quando transportarem TRUE. Isto permite o monitoramento constante do fluxo de informações.

Considerar:

1. tempo de execução de um programa será aumentado pelo uso do controle de fluxo. Isto pode causar timeouts em programas cíclicos com alta carga;
2. Em posições do breakpoint ativo não há nenhuma exibição de controle de fluxo;
3. Se um cão de guarda tiver sido definido para a tarefa referida, ele será desligado quando o controle de fluxo estiver ativo.

Menu ‘Comunicação’ ‘Modo simulação’

Se o **Modo de Simulação** for escolhido, então um sinal de verificação (✓) aparecerá na frente do item no menu.

Neste modo, que é usado para testar o projeto, o programa do usuário é executado no PC (Windows). A comunicação entre o PC e o modo de Simulação usa o mecanismo de mensagem do Windows.

Se o programa não estiver no modo de simulação, ele será executado no CP. A comunicação entre o PC e o CP geralmente é executada através da interface serial.

O status desse sinalizador é armazenado com o projeto.

Considerar: POUs das bibliotecas externas não serão executadas no modo de simulação.

Menu ‘Comunicação’ ‘Parâmetros de comunicação’

Obtém-se uma caixa de diálogo especial para definir os parâmetros de comunicação quando a comunicação entre o PC local e o controlador for executada sobre um servidor de gateway no seu sistema.

Consultar os seguintes itens:

- Princípio de um sistema de gateway;
- Diálogo de parâmetros de comunicação para o PC local;
- Configurando o servidor de Gateway e o canal desejado;
- Configurar um novo canal para o servidor de gateway local;
- O que a caixa de diálogo de parâmetros de comunicação no PC local mostra.

Princípio de um sistema gateway

Vamos examinar o princípio do sistema gateway antes de explicar a operação da caixa de diálogo:

Um servidor gateway pode ser usado para permitir que o seu PC local possa se comunicar com um ou mais controladores. As configurações relativas à quais controladores podem ser endereçados - os quais estão configurados especificamente para cada servidor gateway e a conexão com o servidor de Gateway desejado - são feitas no PC local. Ali é possível que ambos, o servidor gateway e o controlador sejam executados juntos no PC local. Caso esteja-se lidando com um servidor gateway que está sendo executado em outro PC, deve-se garantir que ele tenha sido iniciado lá. Se for selecionando um servidor gateway instalado localmente, ele iniciará automaticamente ao fazer login no controlador do dispositivo. Pode-se reconhecê-lo por meio de um símbolo do **MasterTool IEC** na parte inferior direita na barra de tarefas. Esse símbolo aparecerá enquanto se estiver conectado ao controlador sobre o gateway. Os itens do menu **Informações** e **Finalizar** são obtidos clicando com a tecla direita do mouse no símbolo. **Finalizar** é usado para desativar o gateway.

Veja o seguinte esquema para apresentar um sistema Gateway:

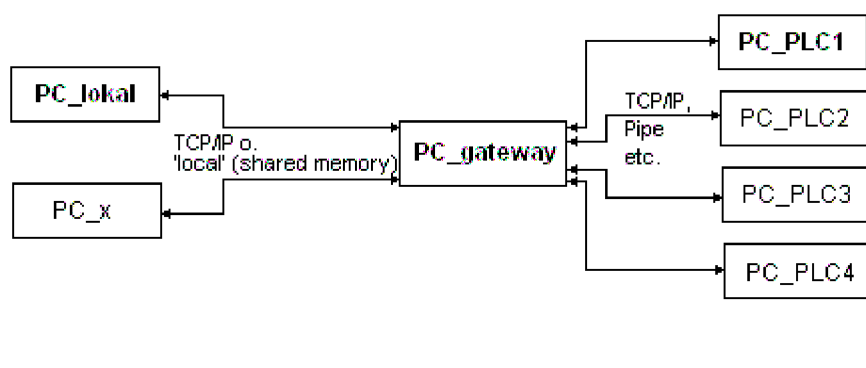


Figura 4-67.

PC_local é o seu PC local; **PC_x** é o outro PC, para o qual o gateway é endereçado. **PC_gateway** é o **PC** no qual o servidor do gateway está instalado, **PC_PLC1** até **PC_PLC4** são PCs onde os controladores estão sendo executados. O diagrama mostra os módulos como separados, mas é totalmente possível que o servidor gateway e/ou controlador sejam instalados em conjunto no PC local.

Importante: por favor, observe que uma conexão para o gateway só é possível sobre TCP/IP, portanto, verifique se seu computador está configurado apropriadamente!

As conexões a partir do gateway para vários computadores, por outro lado, podem ser realizadas em protocolos diferentes (TCP/IP, Pipe, etc.).

Configurando o servidor Gateway e o canal desejado

1. Configurando o servidor de Gateway e o canal desejado na caixa de diálogo de Parâmetros de Comunicação

Para definir a conexão desejada com o servidor Gateway, abra a caixa de diálogo 'Parâmetros de comunicação do Gateway', pressionando o botão de **Gateway**.

Exemplo de diálogo e definição da conexão local para o Gateway:

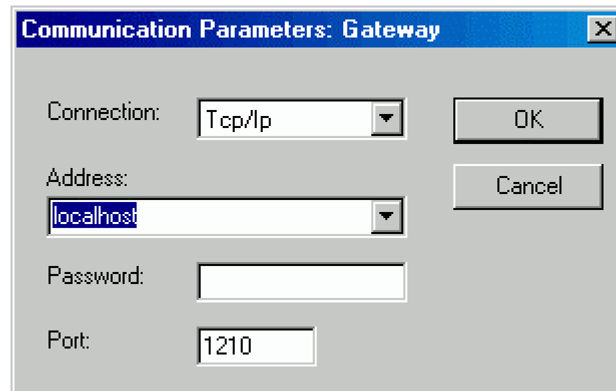


Figura 4-68. Exemplo de diálogo e definição da conexão local para o Gateway

Aqui pode-se inserir e/ou editar o seguinte:

O **tipo** de conexão do computador com o computador no qual o servidor gateway desejado está sendo executado. Se o servidor gateway estiver rodando no computador local, a conexão é possível por meio de memória compartilhada ("local") ou por meio de TCP / IP; se a conexão com um computador diferente for necessária, apenas o TCP / IP pode ser usado.

O **endereço** do computador, no qual o servidor gateway desejado está sendo executado: endereço IP ou o nome simbólico apropriado, como por exemplo, localhost. Considere que não é permitido utilizar zeros à esquerda adicionados ao número de intervalo de endereço (exemplo: não é possível: '010.107.084.050', deve ser inserido como '10.107.84.50'). Na configuração inicial, o padrão 'localhost' é oferecido como o nome do computador (endereço), o que significa que o gateway localmente instalado pode ser acessado. O nome 'localhost' está configurado para ser idêntico ao endereço IP local 127.0.0.1 na maioria dos casos, mas, em alguns casos, é necessário inserir isso diretamente no campo Endereço. Para acessar um servidor gateway em outro computador, deve-se substituir 'localhost' com seu nome ou endereço IP.

A **senha** para o servidor gateway selecionado (se ele estiver em um computador remoto). Se a senha for digitada incorretamente ou não for completamente inserida, uma mensagem de erro será exibida.

Nota: nesta conexão pode ser atribuída ao servidor gateway instalado localmente uma senha com o seguinte procedimento: clique com o botão direito do mouse no símbolo gateway e, na parte inferior direita da barra de ferramentas, selecione 'Alterar senha' . Uma caixa de diálogo aparecerá para alterar ou digitar uma senha. Ao acessar o servidor gateway localmente não haverá necessidade de inserir nenhuma senha .

A **porta** do computador no qual o servidor gateway desejado está em execução. Como regra, o valor correto para o gateway selecionado já é fornecido.

Se a caixa de diálogo for fechada com **OK**, a entrada correspondente (endereço do computador) aparece no campo Canais, na parte superior da caixa de diálogo 'Parâmetros de comunicação' e, abaixo dela, os canais disponíveis neste servidor gateway.

2. Configurando o canal desejado no servidor gateway selecionado

Agora selecione um dos canais, clicando em uma entrada com o mouse. Os parâmetros correspondentes serão mostrados na tabela. Se nenhuma conexão puder ser estabelecida para o endereço de gateway selecionado — possivelmente, porque ele não foi iniciado ou porque o endereço está incorreto — a frase 'Não conectado' aparece entre parênteses após o endereço e uma mensagem 'Nenhum gateway com essas configurações pôde ser encontrado' aparece. Nesta conexão execute uma verificação rápida.

Uma vez que o canal desejado está configurado, feche a caixa de diálogo usando OK. As configurações são salvas com o projeto.

Configurando um novo canal para o servidor Gateway local

Pode-se configurar para o servidor gateway conectado no momento, novos canais que estejam disponíveis para estabelecer mais conexões a partir do servidor (uma conexão com um controlador, por exemplo). As opções disponíveis dependem da opção específica do número de drivers de dispositivo instalados no seu computador.

Pressione o botão **Novo**, na caixa de diálogo de Parâmetros de comunicação. A caixa de diálogo **Parâmetros de Comunicação: Novo Canal** aparece:

Exemplo de Caixa de diálogo Instalando um novo canal

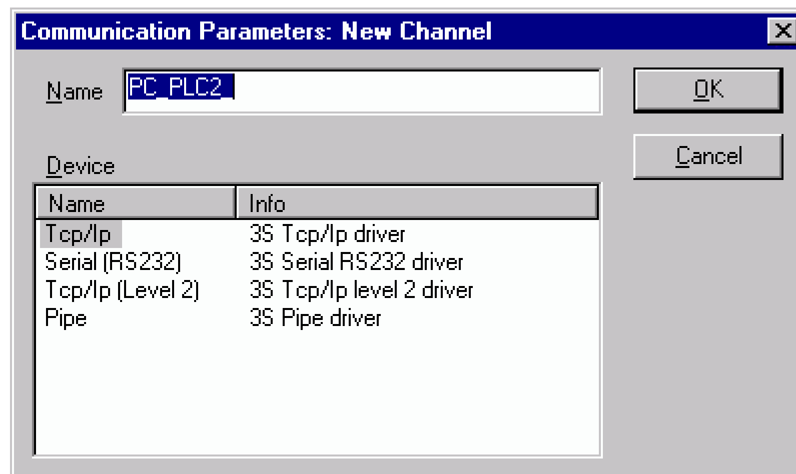


Tabela 4-6. Exemplo de Caixa de diálogo Instalando um novo canal

O campo de entrada **Nome** automaticamente contém o nome usado para o último canal inserido. Se nenhum canal tiver sido definido ainda, o nome do gateway atual será oferecido, seguido por um caractere de sublinhado, por exemplo, 'localhost_'. Pode-se editar o nome do canal nesse momento. O nome do canal é puramente informativo, ele não precisa ser um nome exclusivo, embora seja recomendável usar um.

Os drivers do **dispositivo** disponíveis no computador do gateway são listados na tabela Dispositivos. Na coluna Nome, selecione com um clique do mouse um dos drivers disponíveis. O comentário correspondente, se houver, aparece na coluna de informações.

Ao fechar a caixa de diálogo '.. .Novo canal' com **OK**, o canal recém-definido aparece na caixa de diálogo de 'Parâmetros de comunicação' como uma nova entrada em **Canais** na posição mais baixa sob o sinal de subtração. Até agora, ele foi somente armazenado localmente no projeto (veja acima). Nesse ponto, pode-se editar a coluna de Valor (consultar as dicas abaixo). Agora confirme os parâmetros inseridos com OK, saindo, assim, da caixa de diálogo de Parâmetros de comunicação.

Para que o recém inserido canal gateway e seus parâmetros também sejam conhecidos como servidor gateway xy, e desta forma torne-se disponível para outros computadores que acessam este servidor gateway xy, deve-se efetuar login no controlador. Se, então, a caixa de diálogo 'Comunicação'

'Parâmetros de comunicação' for reaberta, o novo canal aparecerá na "Árvore de canais", não apenas em sua posição anterior, mas também recuado sob o endereço ou nome de dispersão de servidor gateway. Isso indica que ele é conhecido para a rede. Pode-se abrir a caixa de diálogo de Parâmetros de comunicação em um computador diferente do local, selecionar o gateway xy e usar seu novo canal.

Se um erro de comunicação ocorrer ao fazer login, é possível que a interface não possa ser aberta (por exemplo, COM1 para uma conexão serial), possivelmente, porque ela está sendo usada por outro dispositivo. Também é possível que o controlador não esteja em execução.

Os parâmetros para um canal já conhecido pelo servidor gateway não podem ser editados na caixa de diálogo de configuração. Os campos de parâmetros aparecem na cor cinza. No entanto, pode-se excluir a conexão, desde que ela não esteja ativa.

Importante: por favor, note que a exclusão de um canal não é reversível. Ela ocorre no momento que se pressionar o botão **Remover**!

O que mostra a caixa de diálogo dos parâmetros de comunicação no PC local

Esta caixa de diálogo é usada para selecionar um servidor gateway para a comunicação com um CP. Portanto, ali podem ser configurados novos canais para um servidor gateway instalado no PC local para que esses canais possam ser usados por outros computadores que fazem parte da rede.

As configurações atuais podem ser chamadas a qualquer momento usando o botão **Atualizar**.

A caixa de diálogo aparecerá como segue se os parâmetros de comunicação já tiverem sido configurados de acordo com o exemplo em 'Princípios de um sistema de gateway'.

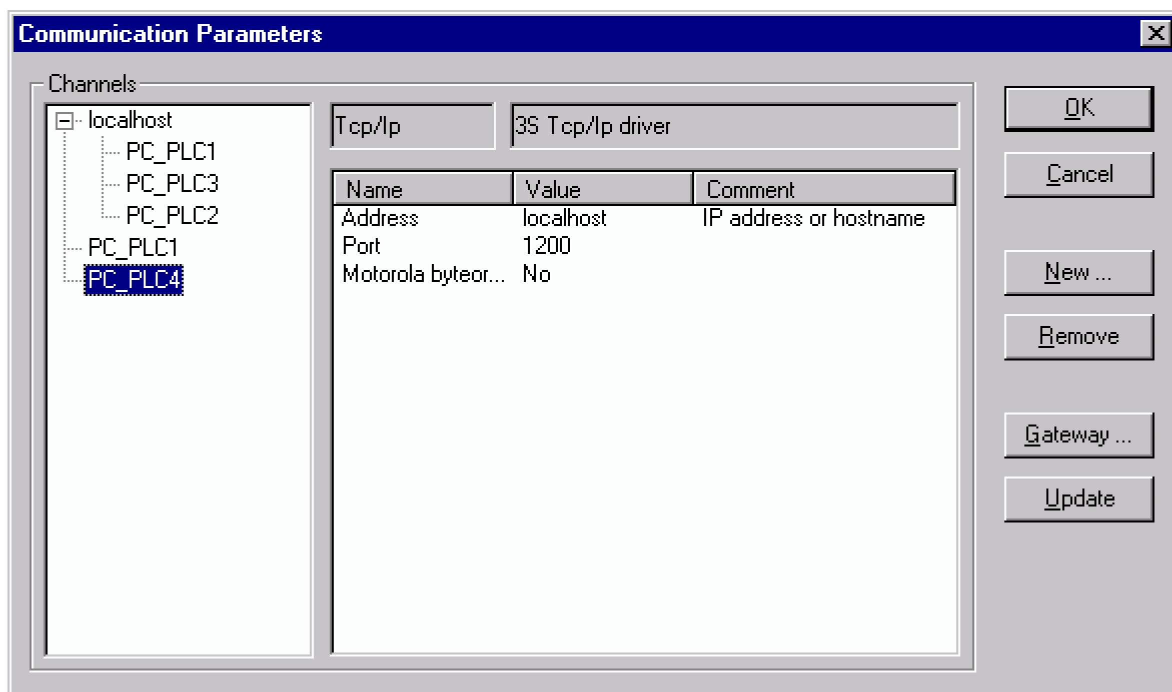


Figura 4-69. Parâmetros de comunicação

Os canais de cabeçalho listam duas categorias de conexões:

Por um lado, todas as conexões instaladas no servidor gateway conectado no momento (chamado 'localhost') são mostradas. Aqui o endereço ou o nome deste gateway está localizado em posição superior atrás do sinal de subtração, que, no nosso exemplo, está em execução no computador local. O endereço apropriado 'localhost' corresponde, normalmente, ao endereço IP 127.0.0.1 do computador local (PC_local). A seguir, recuados para a direita, estão três endereços de computadores de tempo de execução para os quais os canais de gateway são configurados (PC_PLC1 para 3). Eles

poderiam ser configurados tanto apartir do PC local quanto através de outros computadores (PC_x) que são ou foram conectados ao servidor gateway.

A segunda categoria dos canais inclui todas as conexões para o gateway que podem ser configuradas no PC local nesta caixa de diálogo de configuração, por exemplo. Eles criam a "ramificação" que leva a partir do sinal de subtração diretamente abaixo de PC_PLC1 e PC_PLC4. Esses endereços de canal não necessariamente precisarão ser conhecidos ainda no gateway. Para PC_PLC4 no exemplo descrito acima, os parâmetros de configuração são armazenados localmente no projeto, mas eles primeiro serão identificados pelo gateway no próximo login no controlador. Isso já ocorreu para PC_PLC1 desde que o endereço de gateway associado apareceu como uma "sub-ramificação" adicional à "Árvore de canais".

Na parte central da caixa de diálogo localiza-se a designação, em cada caso, do canal selecionado à esquerda e o parâmetro associado sob o Nome, Valor e Comentários.

Dicas para editar os parâmetros no diálogo dos parâmetros de comunicação

Somente podem-se editar os campos de texto na coluna **Valor**.

Selecione um campo de texto com o mouse e entre no modo de edição clicando duas vezes ou pressionando a barra de espaço. A entrada de texto é concluída, pressionando a tecla <Enter>.

Pode-se usar <Tab> ou <Shift>+<Tab> para saltar para a possibilidade de edição ou seleção seguinte (ou anterior).

Para editar valores numéricos é possível utilizar as setas de direção ou as teclas Page Up/Down, visando alterar o valor por uma ou dez unidades respectivamente. Um clique duplo com o mouse também altera o valor, aumentando uma unidade. Uma verificação de digitação é instalada para valores numéricos:<Ctrl>+<Home> ou <Ctrl>+<End> trazem o menor ou o maior valor respectivamente para os possíveis valores de entrada para o tipo de parâmetro em questão.

Verificação rápida no caso de tentativa de conexão sem êxito para o gateway

Deve-se fazer as seguintes verificações se a conexão com o computador gateway selecionado não for bem sucedida, (aparecerá a mensagem de não conexão na caixa de diálogo de Parâmetros de comunicação atrás do endereço do servidor gateway no campo relativo aos canais):

- O servidor gateway foi iniciado (o símbolo de três cores aparece na parte inferior direita da barra de ferramentas);
- O endereço IP que inserido no diálogo 'Gateway: parâmetros de comunicação' é realmente o endereço do computador no qual o gateway está sendo executado? (Use “ping" para verificar);
- A conexão de TCP/IP está funcionando localmente? O erro, possivelmente, deve estar com o TCP/IP.

Menu ‘Comunicação’ ‘Download de código fonte’

Este comando carrega o código fonte para o projeto no sistema do controlador. Isso não deve ser confundido com o código criado quando o projeto é compilado! Pode-se inserir as opções que se aplicam ao download (hora, tamanho) na caixa de diálogo 'Projeto' 'Opções' 'Download de fonte'.

Menu ‘Comunicação’ ‘Criar projeto de inicialização’

Com este comando, o projeto compilado é configurado no controlador de tal forma que este poderá carregá-lo automaticamente quando reiniciado. O armazenamento do projeto de inicialização ocorre de diferentes formas, dependendo do sistema do dispositivo. Por exemplo, nos sistemas 386, três arquivos são criados: **default.prg,** que contém o código do projeto, **default.chk,** que contém o checksum do código e **default.sts** ,que contém o status do controlador após a reinicialização (iniciar/parar).

O comando 'Comunicação' 'Criar projeto de inicialização' também está disponível no **modo offline,** se o projeto foi compilado sem erros. Nesse caso, os seguintes arquivos são criados no diretório dos

projetos: <**nome do projeto**>**.prg (**para o código do projeto de inicialização) e <**nome do projeto**>**.chk** (para o checksum). Esses arquivos podem ser renomeados conforme necessário e então, copiados para um CP.

Dependendo das configurações do sistema do dispositivo, na criação de um projeto de inicialização no modo offline, talvez seja criado um novo arquivo *.ri (informações de download). Também dependendo do dispositivo, uma caixa de diálogo será exibida (se este arquivo já existir).

Nota: se a opção do projeto **Implícito ao criar projeto de inicialização** (categoria download de fonte) estiver ativada, então as fontes selecionadas serão carregadas automaticamente para o controlador no comando 'Comunicação' 'Criar projeto de inicialização'.

Menu 'Comunicação' 'Enviar arquivo para o CP'

Este comando é usado para enviar qualquer arquivo desejado para o controlador. Ele abre a caixa de diálogo 'Enviar arquivo para o CP' na qual pode-se selecionar o arquivo desejado.

Depois que a caixa de diálogo é fechada com o botão 'Abrir', o arquivo é enviado para o controlador e armazenado lá sob o mesmo nome. O processo de envio é acompanhado por uma caixa de diálogo de progresso.

Com o comando 'Comunicação' 'Carregar arquivo a partir do CP', pode-se recuperar um arquivo carregado anteriormente no controlador.

Menu 'Comunicação' 'Ler arquivo do CP'

Com este comando, pode-se recuperar um arquivo carregado anteriormente no controlador quando selecionou a opção 'Comunicação' Enviar arquivo para CP'. Obtém-se a caixa de diálogo 'Carregar arquivo a partir do controlador'. No campo nome do arquivo forneça o nome do arquivo desejado e, na janela de seleção, insira o diretório do seu computador no qual ele deverá ser carregado, assim que a caixa de diálogo for fechada com o botão "Salvar".

Configurações de janelas

No item do menu "**Janela**" encontram-se todos os comandos para controlar as janelas. Existem comandos para a configuração automática de sua janela,comandos para a abertura do gerenciador de bibliotecas, e também comandos para alternar entre as janelas abertas.

Na extremidade do menu encontra-se uma lista de todas as janelas abertas, na seqüência em que foram abertas. Pode-se ir para a janela desejada clicando o mouse na entrada respectiva. Um sinal aparecerá na frente da janela ativa.

Menu 'Janela' 'Ajuste horizontal'

Com este comando pode-se organizar horizontalmente todas as janelas na área de trabalho, para que elas não sejam sobrepostas e não preencham a área de trabalho inteira.

Menu 'Janela' 'Ajuste vertical'

Com este comando pode-se organizar verticalmente todas as janelas na área de trabalho, para que elas não sejam sobrepostas e não preencham a área de trabalho inteira.

Menu 'Janela' 'Cascata'

Com este comando pode-se organizar todas as janelas na área de trabalho em forma de cascata, uma atrás da outra.

Menu ‘Janela’ ‘Ordenar símbolos’

Com este comando pode-se organizar todas as janelas minimizadas na área de trabalho em uma fileira na extremidade inferior da área de trabalho.

Menu ‘Janela’ ‘Fechar tudo’

Com este comando pode-se fechar todas as janelas abertas na área de trabalho.

Menu ‘Janela’ ‘Mensagens’

Atalho:<Shift>+<Esc>

Com este comando pode-se abrir ou fechar a janela de mensagens com as mensagens da última compilação, verificação ou procedimento de comparação

Se a janela de mensagens for aberta, uma seleção (✓) aparecerá na frente do comando.

A ajuda

Ajuda: conteúdo e pesquisa

Com os comandos 'Conteúdo e Pesquisa' no menu Ajuda, pode-se abrir a janela Tópicos de ajuda, que será exibida através do Visualizador de ajuda HTML (Internet Explorer V4.1 e posteriores).

A guia **Conteúdos** mostra a árvore de conteúdos. Os livros podem ser abertos e fechados através de um clique duplo ou nos sinais de adição e subtração. A página que está atualmente selecionada na árvore de conteúdos será exibida na parte direita da janela. Hiperlinks do texto para outras páginas de ajuda e hotspots expandidos são marcados por uma cor diferente e um sublinhado. Um clique do mouse em tais textos abrirá a página vinculada e mostrará o texto expandido ou uma imagem. Por exemplo, pode-se clicar em "Janela de tópico de ajuda" no final dessa página para obter uma imagem da Janela da Ajuda, ou pode-se clicar no hiperlink "Ajuda sensível ao contexto" para ir para esta página respectiva de ajuda.

Na guia **Índice** pode-se procurar páginas de ajuda em itens específicos; na guia **Pesquisar** pode ser feita uma pesquisa completa de textos em todas as páginas de ajuda. Siga as instruções nos cartões de registro.

Janela Tópicos de ajuda:

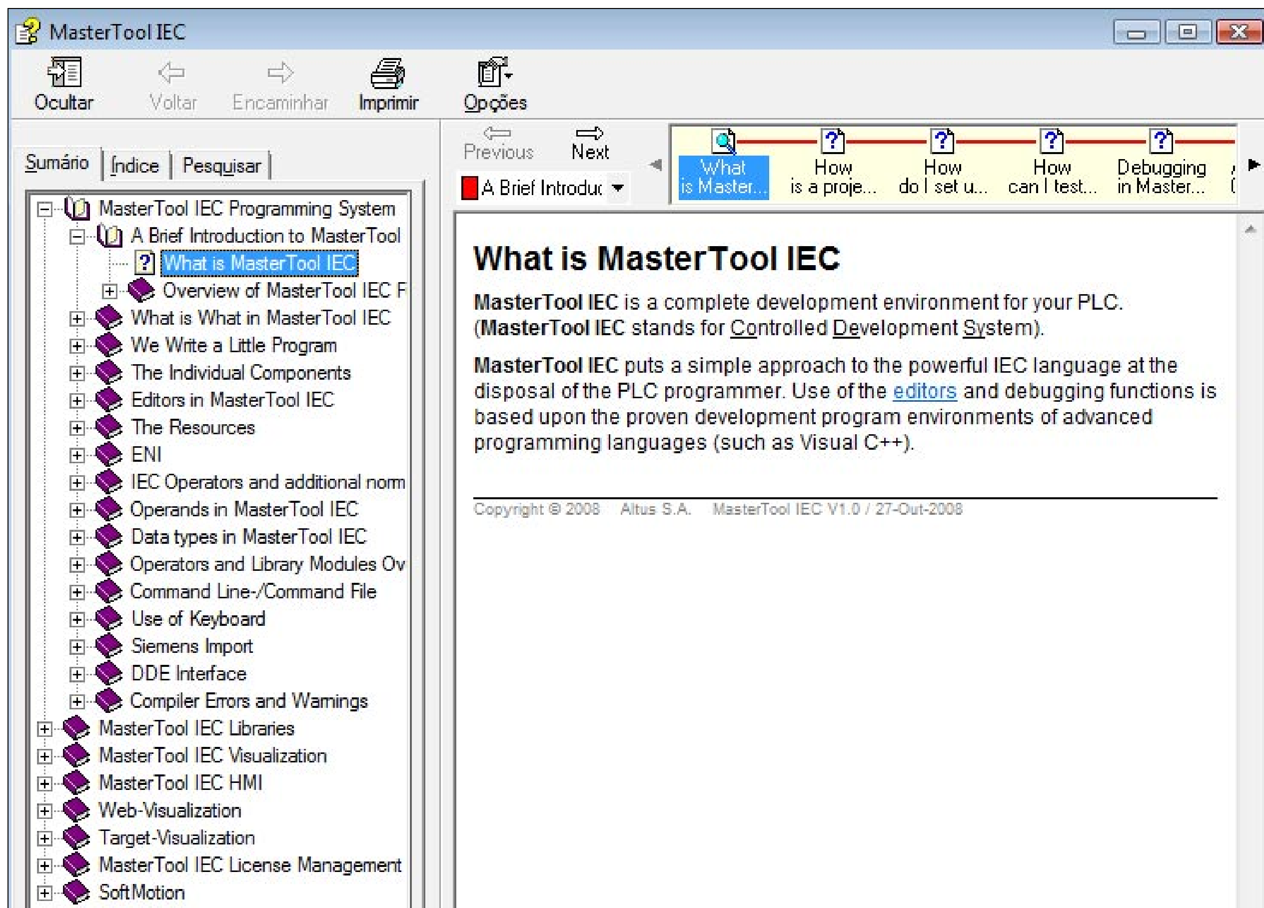


Figura 4-70. Janela Tópicos de ajuda

Ajuda sensível ao contexto

Atalho: <F1>

Pode-se usar a tecla <F1> em uma janela ativa, em uma caixa de diálogo ou no comando do menu para abrir a ajuda online. Ao executar o comando a partir do menu, será exibida a ajuda para o comando chamado n. o momento

Também pode-se realçar um texto (por exemplo, uma palavra-chave ou uma função padrão) e pressionar <F1> para que seja exibida a ajuda para esse item.

5. Editores no MasterTool IEC

Características gerais dos editores

Componentes de um editor

Todos os editores para POUs (Unidades de Organização de Programa) consistem em uma parte de declaração e um corpo. O corpo pode consistir de um texto ou de um editor gráfico; e a parte da declaração é sempre um editor de texto. Corpo e Declaração são separados por um divisor de tela que pode ser arrastado, conforme solicitado, clicando com o mouse e movendo-o para cima ou para baixo.

Margens de impressão

As margens verticais e horizontais que se aplicam quando o conteúdo do editor é impresso, são mostradas por linhas tracejadas vermelhas se a opção **'Mostrar intervalo de impressão'** nas opções de projeto na caixa de diálogo **'Área de trabalho'** foi selecionada. As propriedades da impressora, bem como o tamanho do leiaute de impressão são aplicadas através do menu 'Arquivo' 'Configurar impressora'. Se não foram definidas nenhuma configuração de impressora e nenhum leiaute de impressão, será usada uma configuração padrão (Default. DFR e impressora padrão). As margens horizontais são desenhadas como se as opções 'Nova página para cada objeto' ou 'Nova página para cada sub-objeto' fossem selecionada nas configurações da documentação. A margem inferior não é exibida.

Nota: uma exibição exata das margens de impressão só é possível quando o fator de zoom é selecionado em 100 %.

Comentários

Comentários do usuário devem estar entre seqüências de símbolos especiais '*' . Exemplo: (*Este é um comentário.*)

Comentários são permitidos em todos os editores de texto, em qualquer local desejado, isto é, em todas as declarações, nas linguagens IL e ST e nos tipos de dados auto-definidos. Se o projeto é impresso usando um **modelo**, o comentário inserido durante a declaração de variáveis será exibido em componentes de programa com base no texto após cada variável.

Nos editores gráficos FBD e LD, comentários podem ser inseridos para cada rede. Para fazer isso, selecione a rede para inserir o comentário e ative **'Inserir' 'Comentário'**.

Além disso, comentários sempre podem ser adicionados onde nomes de variáveis são inseridos.

Exemplo no FBD para um comentário de rede e para um comentário colocado atrás de uma variável de entrada:

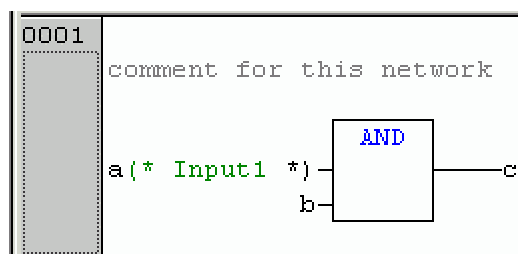


Figura 5-1. Exemplo de comentários em FBD

No Editor LD, também pode ser adicionado um comentário para cada contato e para cada bobina, se as opções correspondentes estiverem ativadas no menu 'Extras' 'Opções'.

No CFC existe um comentário especial de POUs.

No SFC, pode-se inserir comentários sobre um passo na caixa de diálogo para editar atributos de passo.

Comentários aninhados também são permitidos se a opção apropriada na caixa de diálogo 'Projeto' 'Opções de compilação' estiver ativada.

No modo online, ao posicionar o cursor do mouse por um curto período de tempo em uma variável, o tipo e, se aplicável, o endereço e o comentário dessa variável são exibidos em uma tooltip.

Zoom para atalho de POU

Atalho: <Alt>+<Enter>

Com este comando uma POU selecionada é carregada em seu editor. O comando está disponível no menu de contexto (<F2>). Também estará disponível no menu 'Extras' se o cursor estiver posicionado sobre o nome de uma POU (em um editor de texto) ou se a caixa POU estiver selecionada (em um editor gráfico).

Ao lidar com a POU de uma biblioteca, então o gerenciador de bibliotecas será chamado e a POU correspondente será exibida.

Abrir instância

Este comando corresponde ao comando 'Projeto' 'Abrir Instância'.

Ele está disponível no menu de contexto (<F2>) ou no menu 'Extras', se o cursor estiver posicionado sobre o nome de um bloco funcional (em um editor de texto) ou se a caixa de bloco funcional estiver selecionada (em um editor gráfico).

Função “Intellisense”

Se a opção "Componentes da lista" estiver ativada na caixa de diálogo de opções do projeto na categoria 'Editor', a funcionalidade "IntelliSense" estará disponível em todos os editores, nos gerenciadores específicos, na Visualização e no Rastreamento de Amostragem:

Ao inserir um ponto ".", em vez de um identificador, será exibida uma caixa de seleção, listando todas as variáveis locais e globais do projeto. Pode-se escolher um destes elementos e pressionar 'Retornar' para inseri-lo atrás do ponto. Também pode-se inserir o elemento via duplo clique na entrada da lista.

Ao inserir uma instância de bloco funcional ou uma variável de estrutura seguida por um ponto, obtém-se uma caixa de seleção listando os componentes da estrutura ou todas as variáveis de entrada/saída do bloco funcional correspondente. Nesse diálogo, poder-se-á escolher e inserir o elemento desejado pressionando a tecla 'Enter' ou executando um clique duplo.

Exemplo: inserir "struvar."-> Os componentes da estrutura serão oferecidos:

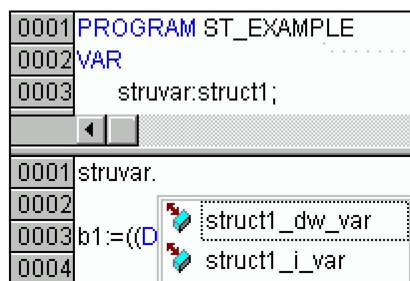


Figura 5-2. Componentes da estrutura ao inserir "struvar"

Ao inserir qualquer string e pressionar <Ctrl>+<Barra de Espaço>, será exibida uma caixa de seleção listando todas as POUs e variáveis globais disponíveis no projeto. A entrada da lista, começando pela string especificada, será selecionada e pode ser inserida no programa, pressionando a tecla <Enter>.

Tooltips no modo offline para identificadores

Se o cursor está colocado em um identificador editável, o nome do identificador, a classe da variável (por exemplo, VAR_GLOBAL), os tipos de dados, os atributos da variável (por exemplo, RETAIN), os endereço e os comentários serão exibidos em uma tooltip. Isto ocorre com todos os editores no modo offline.

Editor de declaração

Trabalhando no editor de declaração

O editor de declaração é usado para declarar variáveis de POUs e variáveis globais, para declarações de tipo de dados e no Gerenciador de Monitoração e Receitas. Ele oferece acesso às funções usuais do Windows, e, se o driver correspondente estiver instalado até mesmo as funções do Intellimouse podem ser usadas.

No modo Sobrescrever (overwrite), '**OV**' é mostrado em preto na barra de status e pode-se alternar entre os modos Sobrescrever e Inserir através da tecla <Ins>.

A declaração de variáveis é suportada pela "cor da sintaxe".

Os comandos mais importantes são encontrados no menu de contexto (botão direito do mouse ou <Ctrl>+<F10>).

Dica: considere a possibilidade de usar pragmas para afetar as propriedades de uma variável no que se refere ao processo de compilação e pré-compilação.

Parte declarações

Todas as variáveis a serem usadas somente nesta POU são declaradas na parte de declarações da POU. Elas podem incluir: variáveis de entrada, variáveis de saída, variáveis de entrada/saída, variáveis locais, variáveis retentivas e constantes. A sintaxe de declaração é baseada na norma IEC61131-3.

Para uma recomendação sobre o uso de identificadores consultar: Apêndice J.

Considere a possibilidade de usar modelos de objetos do tipo 'Variáveis Globais', 'Tipos de dados', 'Função', 'Bloco funcional' ou 'Programa'. Consultar o capítulo 4: 'Arquivo' Novo a partir do modelo'.

Dica: considere a possibilidade de usar pragmas para afetar as propriedades de uma variável no que se refere ao processo de pré-compilação e compilação.

Exemplo de uma declaração correta de variáveis no Editor **MasterTool IEC**:

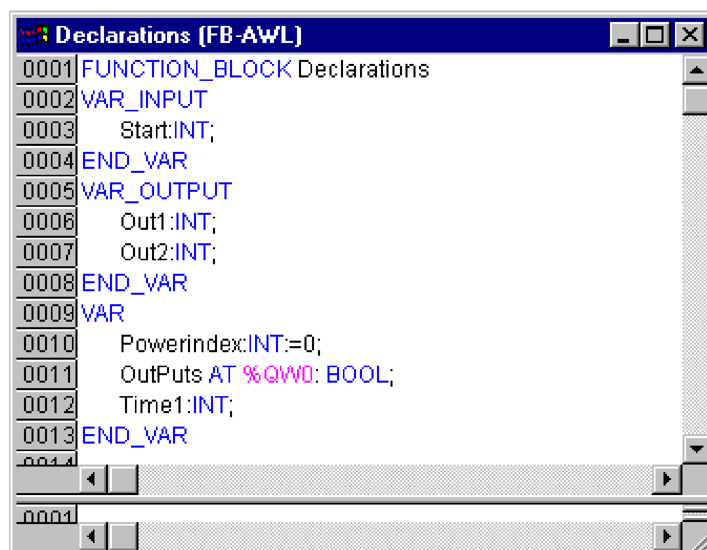


Figura 5-3. Exemplo de uma declaração correta de variáveis no Editor MasterTool IEC

Variável de entrada

Entre as palavras-chave **VAR_INPUT** e **END_VAR**, todas as variáveis que servem como variáveis de entrada para uma POU são declaradas. Isso significa que na posição de chamada, o valor das variáveis poderá ser fornecido juntamente com uma chamada.

Exemplo:

```
VAR_INPUT

IIN1: INT (* 1. VARIÁVEL DE ENTRADA*)

END_VAR
```

Variável de saída

Entre as palavras-chave **VAR_OUTPUT** e **END_VAR**, todas as variáveis que servem como variáveis de saída de uma POU são declaradas. Isso significa que estes valores são retornados para a POU que fez a chamada. Elas podem ser respondidas e usadas posteriormente.

Exemplo:

```
VAR_OUTPUT

IOUT1: INT; (* 1. VARIÁVEL DE SAÍDA*)

END_VAR
```

Variáveis de Entrada e Saída

Entre as palavras-chave **VAR_IN_OUT** e **END_VAR**, todas as variáveis que servem como variáveis de entrada e saída para uma POU são declaradas.

Atenção: com essa variável, o valor da variável transferido é alterado ("transferidas como um ponteiro", chamada por referência). Isso significa que o valor de entrada para essas variáveis não pode ser uma constante. Por este motivo, até mesmo as variáveis de VAR_IN_OUT de um bloco funcional não podem ser lidas ou gravadas diretamente a partir de fora via, <instânciadoblocofuncional><variávelE/S>.

Exemplo:

```
VAR_IN_OUT
```

IINOUT1 :INT;(* 1. VARIÁVEL DE ENTARDA E SAÍDA *)

END_VAR

Variáveis locais

Entre as palavras-chave **VAR e END_VAR**, todas as variáveis locais de uma POU são declaradas. Elas não têm nenhuma conexão externa, ou seja, elas não podem ser gravadas externamente.

Exemplo:

VAR

ILOC1: INT; (* 1. VARIÁVEL LOCAL *)

END_VAR

Variáveis Remanentes

Variáveis remanentes podem manter seus valores durante o período usual de execução do programa. Elas incluem variáveis retentivas e variáveis persistentes.

Exemplo: Variáveis remanentes

VAR RETAIN

IREM1: INT; (* 1. VARIÁVEL REMANENTE*)

END_VAR

Variáveis Retentivas são identificadas pela palavra-chave **RETAIN**. Estas variáveis mantêm seu valor mesmo após um desligamento não controlado do controlador bem como após um desligamento e re-ligamento do controlador (correspondente ao comando 'Comunicação' 'Reset', consultar o capítulo 4). Quando o programa é executado novamente, os valores armazenados serão processados posteriormente. Um exemplo concreto seria um contador de peças em uma linha de produção que recomeça a contagem após uma falha de energia. Todas as outras variáveis são inicializadas, com seus valores inicializados ou com as inicializações padrão. As retentivas são reinicializadas em 'Reset (frio)' e 'Reset (original)' e - ao contrário das Variáveis Persistentes - em um novo download do programa.

Variáveis Persistentes são identificadas pela palavra-chave **PERSISTENT**. Diferentemente das retentivas, essas variáveis mantêm seus valores somente após um novo download, mas não depois de um 'Comunicação' 'Reset', 'Comunicação' 'Reset (original)' ou 'Comunicação' 'Reset (frio)',porque eles não são salvos na "área de retenção" (consultar o capítulo 4). Se as variáveis persistentes também deverem manter seus valores após um desligamento não controlado do controlador, elas precisam ser declaradas adicionalmente como variáveis retentivas. Um exemplo concreto de "variáveis persistentes - retentivas" seria um temporizador de operações que reinicia a contagem do tempo após uma falha de energia.

Atenção: Se uma variável local em um programa for declarada como VAR RETAIN, então, exatamente esta variável será salva na área de retenção (como uma variável retentiva global). Se uma variável local em um **bloco funcional** for declarada como VAR RETAIN, então, a instância completa deste será salva na área de retenção (todos os dados da POU), na qual somente a variável retentiva declarada será tratada como retentiva. Se uma variável local em uma **função** é declarada como VAR RETAIN, então, isso será sem efeito. A variável não será salva na área de retenção! Se uma variável local é declarada como PERSISTENTE em uma função, então, isso será **sem efeito** também!

x = valor será mantido **-** = valor é reinicializado

Após comando online	VAR	VAR RETAIN	VAR PERSISTENT	VAR RETAIN PERSISTENT VAR PERSISTENT RETAIN
Reset	-	x	-	X
Reset (frio)	-	-	-	-
Reset (original)	-	-	-	-
Download	-	-	x	X
Alterações Online	x	x	x	X

Tabela 5-1. Variáveis

Constantes, literais digitadas

Constantes são identificadas pela palavra-chave **CONSTANT**. Elas podem ser declaradas local ou globalmente.

Sintaxe:

```
VAR CONSTANT
<IDENTIFIER>:<TYPE> := <INITIALIZATION>;
END_VAR
```

EXEMPLO:

```
VAR CONSTANT

C_ICON1: INT:=12; (* 1. CONSTANT*)

END_VAR
```

Consultar o Apêndice B: Operandos no MasterTool IEC para obter uma lista de constantes possíveis. Consultar lá também sobre a possibilidade de usar constantes digitadas (Literais Digitadas).

Variáveis externas

Variáveis globais que devem ser importadas para a POU são designadas com a palavra-chave EXTERNAL. Elas também aparecem na janela de exibição específica da parte de declarações no modo online.

Se a declaração VAR_EXTERNAL não coincidir com a declaração global em cada aspecto, aparecerá a seguinte mensagem de erro: "Declaração de '<var>' não corresponde à declaração global!"

Se a variável global não existir, a seguinte mensagem de erro aparecerá: "Variável global desconhecida: '<var>'!"

Exemplo:

```
VAR EXTERNAL
IVAREXT1: INT:=12; (* PRIMEIRA VARIÁVEL EXTERNA *)
END_VAR
```

Palavras-chave

Palavras-chave são escritas em letras maiúsculas em todos os editores. Palavras-chave não podem ser usadas como variáveis. Exemplos de palavras-chave: VAR, VAR_CONSTANT, IF, NOT, INT.

Declaração de variáveis

Uma declaração de variáveis tem a seguinte sintaxe:

<Identificador> {AT <Endereço>}:<Tipo> {: = <inicialização>};

As partes entre chaves {} são opcionais.

Sobre o identificador, que é o nome de uma variável, deve-se observar que ele não pode conter espaços ou caracteres não permitidos. Ele não pode ser declarado em dobro e não pode ser idêntico a qualquer outra palavra-chave. A escrita de variáveis em maiúsculas e minúsculas é ignorada, ou seja, VAR1, Var1 e var1 não são variáveis diferentes. Caracteres sublinhados em identificadores são significativos, por exemplo: A_BCD e AB_CD são interpretados como identificadores diferentes. Vários sublinhados consecutivos no início de um identificador ou dentro de um identificador não são permitidos. O comprimento do identificador, bem como a sua parte significativa, são ilimitados.

Todas as declarações de variáveis e elementos de tipo de dados podem incluir a inicialização. Eles são trazidos pelo operador ":=". Para variáveis de tipos elementares, essas inicializações são constantes. A inicialização padrão é 0 (zero) para todas as declarações.

Exemplo:

```
IVAR1: INT:=12; (* VARIÁVEL INTEGRAL COM VALOR INICIAL DE 12*)
```

Para vincular uma variável diretamente a um endereço definitivo, então deve-se declarar a variável com a palavra-chave **AT**.

Para uma entrada mais rápida das declarações, use o modo de atalho.

Em blocos funcionais também pode-se especificar as variáveis com instruções de endereço incompleto. Para que tal variável seja usada em uma instância local, deve haver uma entrada para ela na configuração da variável.

Preste atenção a possibilidade de uma declaração automática.

Dica: Considere a possibilidade de usar pragmas para afetar as propriedades de uma variável no que se refere ao processo de compilação e pré-compilação.

Declaração AT

Para vincular uma variável diretamente a um endereço definitivo, então deve-se declarar a variável com a palavra-chave **AT**. A vantagem de tal procedimento é que pode-se atribuir um nome significativo para o endereço e as alterações necessárias de um sinal de entrada ou saída só precisarão ser feitas em um lugar (por exemplo, na declaração).

Observe que as variáveis que exigem uma entrada não podem ser acessadas por escrita.

Exemplos:

```
XCOUNTERHEAT7 AT %QX0.0: BOOL;
WLIGHTCABINETIMPULSE AT %IW2: WORD;
XDOWNLOAD AT %MX2.2: BOOL;
```

Nota: se variáveis booleanas forem atribuídas aos endereços de Byte, Word ou DWORD, elas ocuparão um byte com TRUE ou FALSE, não apenas o primeiro bit após o offset!

'Inserir' 'Declaração de palavras-chave'

Pode-se usar este comando para abrir uma lista de todas as palavras-chave que podem ser usadas na parte de declaração de uma POU. Depois que uma palavra-chave for escolhida e a opção tiver sido confirmada, a palavra será inserida na posição atual do cursor.

Também se obtém esta lista ao abrir o Assistente de entrada (<F2>) e escolher a categoria **Declarações**.

'Inserir' 'Tipo'

Com este comando, aparecerá uma seleção de tipos possíveis para uma declaração de variáveis. Esta lista também pode ser obtida quando se acessa o Assistente de entrada (<F2>).

Os tipos são divididos nestas categorias:

- Tipos padrão BOOL, BYTE, etc.;
- Estruturas de tipos definidos, tipos enumeração, etc.;
- Blocos funcionais padrão para declarações de instância;
- Blocos funcionais definidos para declarações de instância.

O **MasterTool IEC** dá suporte a todos os tipos padrão da IEC1131-3:

Cor da sintaxe

Em todos os editores existe suporte visual na implementação e na declaração de variáveis. Dessa forma, os erros são evitados, ou, pelo menos, descobertos mais rapidamente, pois o texto é exibido em cores.

Um comentário deixado em aberto e outros problemas serão percebidos imediatamente; palavras-chave não incorrerão em erros acidentalmente, etc..

As seguintes cores de realce serão usadas:

Azul	**Palavras-chave**
Verde	Comentários em editores de texto
Rosa	Constantes especiais (por exemplo, TRUE/FALSE, T#3s, %IX0.0)
Vermelho	Erro de entrada (por exemplo, constante de tempo inválida, palavra-chave escrita em minúsculas,...)
Preto	Variáveis, constantes, operadores de atribuição...

Tabela 5-2. Cores de realce

Modo de Atalho

O editor de declaração para o MasterTool IEC permite que se use o modo de atalho. Esse modo está ativado quando se encerra uma linha com <Ctrl><Enter>.

Os atalhos a seguir são suportados.

Todos os identificadores até o último identificador de uma linha se tornarão declaração de identificadores de variável.

O tipo de declaração é determinado pelo último identificador da linha. Neste contexto, o seguinte se aplica:

B ou BOOL	gera o resultado	BOOL
I ou INT	gera o resultado	INT
R ou REAL	gera o resultado	REAL
S ou string	gera o resultado	STRING

Tabela 5-3. Atalhos

Se nenhum tipo tiver sido estabelecido através destas regras, o tipo será BOOL e o último identificador não será usado como um tipo (exemplo 1).

Toda constante, dependendo do tipo de declaração, se transformará em uma inicialização ou uma seqüência de caracteres (exemplos 2. e 3.).

Um endereço (como em % MD12) é estendido ao redor do ATATDeclaration>Proc...(atributo) (exemplo 4.).

Um texto após um ponto-e-vírgula (;) torna-se um comentário (exemplo 4.).

Todos os outros caracteres na linha são ignorados (por exemplo, o ponto de exclamação no exemplo5.).

Exemplos:

Atalho	Declaração
A	A: BOOL;
A B I 2	A , B: INT := 2;
ST S 2; A string	ST: STRING(2); (* A string *)
X %MD12 R 5 Real Nr.	X AT %MD12: REAL := 5.0;(* Real Number *)
B !	B: BOOL;

Tabela 5-4. Exemplos de declaração

Auto-declaração

Se a opção Auto-declaração tiver sido escolhida na categoria Editor da caixa de diálogo Opções, então uma caixa de diálogo será exibida em todos os editores após a entrada de uma variável ainda não declarada. Com a ajuda deste diálogo, a variável agora poderá ser declarada.

Caixa de diálogo para declarações de variáveis

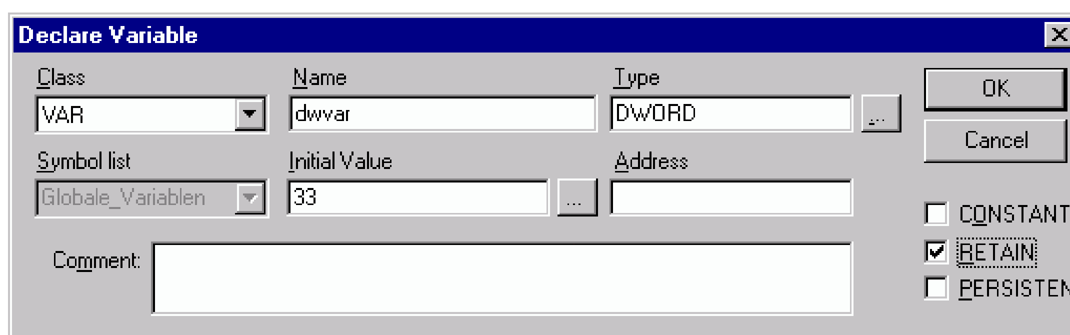


Figura 5-4. Caixa de diálogo para declarações de variáveis

Através da caixa de combinação **Classe**, escolha entre uma variável local (**VAR**), uma variável de entrada (**VAR_INPUT**), uma variável de saída (**VAR_OUTPUT**), uma variável de entrada/saída (**VAR_INOUT**) ou uma variável global (**VAR_GLOBAL**).

Com as opções **CONSTANTE, RETENTIVA, PERSISTENTE**, define-se se está lidando com uma constante ou uma variável retentiva.

O nome de variável digitado no editor foi inserido no campo **Nome** e BOOL foi colocado no campo **Tipo**. Um botão específico abre a caixa de diálogo Assistente de entrada, a qual permite selecionar todos os tipos de dados possíveis.

Declaração de MATRIZ (ARRAY)

Se MATRIZ for escolhida como o tipo de variável, a caixa de diálogo para inserir os seus limites é exibida.

Caixa de diálogo para determinar os limites de matriz durante a declaração automática

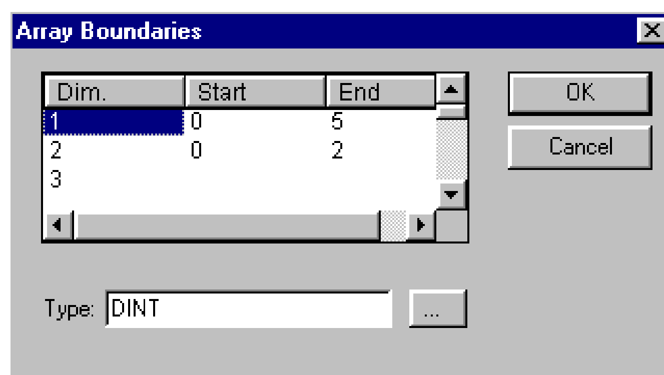


Figura 5-5. Caixa de diálogo para determinar os limites de matriz durante a declaração automática

Para cada uma das três dimensões possíveis (**Dim**.), os limites da matriz podem ser inseridos em **Início** e **Fim** clicando com o mouse no campo correspondente para abrir um espaço de edição. O tipo de dados da matriz é inserido no campo **Tipo**. Ao fazer isso, o botão pode ser usado para chamar uma caixa de diálogo Assistente de entrada.

Após sair da caixa de diálogo de limites da matriz através do botão **OK**, são configuradas declarações de variáveis em formato IEC com base nas entradas no campo Tipo na caixa de diálogo. Exemplo: ARRAY [1..5, 1..3] OF INT

No campo **Valor inicial** pode-se inserir o valor inicial da variável que está sendo declarada. Se esta for uma matriz ou uma estrutura válida, pode-se abrir uma caixa de diálogo de inicialização especial por meio de botão ou abrir a caixa de diálogo do Assistente de entrada para outros tipos de variáveis.

Na caixa de diálogo de inicialização para uma matriz aparecerá uma lista de elementos da matriz; um clique no mouse no seguinte espaço ":=" abre um campo de edição para inserir o valor inicial de um elemento.

Na caixa de diálogo de inicialização para uma estrutura, os componentes individuais são exibidos em uma estrutura de árvore. O tipo e o valor inicial padrão aparecem entre parênteses após o nome da variável seguida de ":=". Um clique no mouse no seguinte campo ":="abre um campo de edição, onde pode-se inserir o valor inicial desejado. Se o componente for uma matriz, então, a exibição de campos individuais na matriz pode ser expandida por um clique do mouse no sinal de mais antes do nome da matriz e os campos podem ser editados com valores iniciais.

Depois de sair da caixa de diálogo de inicialização com **OK**, a inicialização da matriz ou a estrutura aparece no campo Valor inicial da caixa de diálogo de declaração no formato IEC.

Exemplo: x:=5,field:=2,3,struct2:=(a:=2,b:=3)

No campo **Endereço**, pode-se vincular a variável que está sendo declarada a um endereço IEC (declaração AT).

Se aplicável, insira um **comentário**. O comentário pode ser formatado com quebras de linha usando a combinação de teclas<Ctrl><Enter>.

Pressionando **OK**, a caixa de diálogo de declaração é fechada e a variável é inserida no editor de declaração correspondente de acordo a sintaxe IEC.

Nota: a caixa de diálogo para declaração de variável também pode ser acessada pelo comando 'Editar' 'Declarar variável'. Se o cursor está posicionado sobre uma variável no modo online, a janela Auto-declaração pode ser aberta com <Shift><F2>.

Números de linha no editor de declaração

No modo offline, um simples clique em um número de linha especial marcará a linha de texto inteira.

No modo online, um único clique em um número de linha específico irá abrir ou fechar a variável na linha (se uma variável estrutural estiver envolvida).

Declarações como tabelas

Se a opção **Declarações como tabelas** está ativada nas opções de projeto (categoria 'Editor') ou – se estiver trabalhando no editor de declaração – a opção for selecionada no menu de contexto, o editor de declaração se assemelhará a uma tabela. Como em uma caixa de índice de cartão, pode-se selecionar os cartões de registro dos respectivos tipos de variáveis e editá-las.

Para cada variável estão disponíveis os seguintes campos de entrada.

Nome:	**Entrada do identificador da variável**
Endereço:	Se necessário: entrada do endereço da variável (declaração AT)
Tipo:	Entrada do tipo da variável (entrada do bloco funcional ao instanciar o bloco funcional)
Inicial:	Inserir uma inicialização possível da variável (correspondente ao ":= " operador de atribuição).
Comentário:	Inserir um comentário aqui.

Tabela 5-5. Campos de entrada

Ambos os tipos de exibição do editor de declaração podem ser alterados sem causar problemas. No modo online, não há nenhum tipo de exibição diferente.

Inserir nova declaração:

Para editar uma nova variável, selecione o comando 'Inserir' 'Nova Declaração'.

Classificando as declarações

Para classificar as entradas da tabela, posicione o cursor para a barra de número de linha na borda esquerda da janela do editor e escolha um dos seguintes comandos no menu de contexto:

- **Classificar por Nome**: todas as linhas são classificadas em ordem alfabética de acordo com os nomes do identificador na coluna 'Nome';
- **Classificar por Endereço**: todas as linhas são classificadas em ordem alfabética de acordo com as entradas de endereço na coluna 'Endereço';
- **Classificar por Tipo**: todas as linhas são classificadas em ordem alfabética de acordo com os nomes de tipo na coluna 'Tipo';
- **Ordem Original**: as linhas são exibidas na ordem na qual elas foram inseridas originalmente.

Editor de declaração como uma tabela:

PLC_PRG (PRG-FBD)

VAR | VAR_INPUT | VAR_OUTPUT | VAR_IN_OUT | CONSTANT | RETAIN | INI

	Name	Address	Type	Initial	Comment
0001	Iebt		DINT	0	
0002	SG		AWL_EXAMPLE		
0003	Sinus		REAL		
0004	Cosinus		REAL		
0005	r1		REAL		
0006	by1		BYTE		
0007	by2		BYTE		

Figura 5-6. Editor de declaração como uma tabela

'Inserir' 'Nova Declaração'

Com este comando coloca-se uma nova variável na tabela de declaração do editor de declaração. Se a posição atual do cursor estiver localizada em um campo da tabela, então, uma nova variável será colada na linha anterior; caso contrário, a nova variável é colada no final da tabela. Além disso, pode-se colar uma nova declaração no final da tabela usando as teclas de setas para a direita ou a tecla Tab no último campo da tabela.

Aparecerá uma variável com o "Nome" localizado no campo Nome, e "Bool" localizado no campo Tipo, coforme sua configuração padrão. Deve-se alterar estes valores para os valores desejados. Nome e tipo são tudo o que é necessário para uma declaração completa de variáveis.

Editores de declaração no modo online

No modo online, o editor de declaração transforma-se em uma janela de monitoração. Em cada linha há uma variável seguida pelo sinal de igual (=) e pelo valor da variável. Se a variável, neste momento, está indefinida, três pontos de interrogação (?) aparecerão. Para blocos funcionais, os valores são exibidos somente para instâncias abertas (comando: 'Projeto' 'Abrir instância').

Na frente de cada variável multi-elemento há um sinal de adição. Pressionando <Enter> ou após um clique duplo nesta variável, ela será aberta. No exemplo, a estrutura “traffic signal” deve ser aberta.

```
⊟ AMPEL1
   .STATUS = 3
   .GRUEN = FALSE
   .GELB = FALSE
   .ROT = TRUE
   .AUS = FALSE
```

Figura 5-7. Exemplo de abertura de variável

Quando uma variável é aberta, todos os seus componentes são listados depois dela. Um sinal de menos aparece na frente da variável. Ao dar um clique duplo novamente ou pressionar <Enter>, a variável será fechada e o sinal de adição reaparecerá.

Pressionando <Enter> ou com um duplo clique em uma variável de elemento único abrirá a caixa de diálogo para escrever uma variável. Ali é possível alterar o valor presente da variável. No caso de variáveis booleanas, nenhuma caixa de diálogo aparece; essas variáveis são alternadas.

O novo valor será exibido após a variável, na cor turquesa em parênteses pontilhados, e permanecerá inalterado. Se o comando 'Comunicação' 'Escrever valores' for executado, todas as variáveis são colocadas na lista selecionada e são exibidas novamente em preto.

Se o comando 'Comunicação' 'Forçar valores' for executado, então, todas as variáveis serão definidas nos valores selecionados, até que o comando 'Liberar forçamento' seja dado. Nesse caso, a cor do valor forçado muda para vermelho.

Instruções pragma no editor de declaração

A instrução pragma é usada para afetar as propriedades de uma variável no que se refere ao processo de compilação e pré-compilação. Ela pode ser usada com texto suplementar em uma linha de programa do editor de declaração ou em sua própria linha.

A instrução pragma é colocada entre chaves; letras maiúsculas e minúsculas são ignoradas: {<Texto da instrução>}

Se o compilador não puder interpretar o texto de instrução, o pragma inteiro será tratado como um comentário e lido. Um aviso será emitido nesse caso.

Dependendo do tipo e conteúdo do pragma, ele tanto opera na linha na qual ele está localizado quanto em todas as linhas subseqüentes até que ela seja finalizada por um pragma apropriado, ou o mesmo pragma seja executado com parâmetros diferentes, ou ainda, quando o final do arquivo seja atingido. Por arquivo queremos dizer aqui: parte de declarações, parte da implementação, lista de variável global e tipo de declaração.

O parêntese aberto é imediatamente seguido de um nome de variável. Parênteses abertos e fechados devem estar localizados na mesma linha.

Os seguintes pragmas estão atualmente disponíveis no MasterTool IEC:

- Pragma {flag} para Inicialização, Monitoração e Criação de símbolos
- Pragma {bitaccess...} para acesso a bit
- Pragma {link} para vincular POUs durante a compilação
- Pragma {parameter..}, {template...}, {instance...} para criar entradas no Gerenciador de parâmetros
- Pragmas para Controlar a Exibição de Partes de Declaração de Bibliotecas

Pragma {flag} para Inicialização, Monitoração, Criação de símbolos

{FLAG [<FLAGS>] [OFF|ON]}

<flags> pode ser uma combinação dos sinalizadores a seguir:

Noinit	A variável não será inicializada
Nowatch	A variável não pode ser monitorada
Noread	A variável é exportada para o arquivo de símbolos sem permissão de leitura
Nowrite	A variável é exportada para o arquivo de símbolos sem permissão de escrita.
Noread, Nowrite	A variável não será exportada para o arquivo de símbolos.

Tabela 5-6. Pragma {flag}

Com o modificador "**on**" , o pragma opera em todas as declarações de variáveis subseqüentes até que elas sejam encerradas pelo pragma {flag **off**}, ou, até que seja substituído por outro pragma {flag <flags> on}.

Sem o modificador "on" ou "off" , o pragma opera somente na declaração de variável atual (que é a declaração fechada pelo próximo ponto-e-vírgula).

Exemplos para uso de pragma {flag}:

Inicialização e monitoração de variáveis

A variável a não será inicializada e nem monitorada. A variável b não será inicializada:

```
VAR
 A : INT {FLAG NOINIT, NOWATCH};
 B : INT {FLAG NOINIT };
END_VAR
VAR
 {FLAG NOINIT, NOWATCH ON}
```

```
 A : INT;
 {FLAG NOINIT ON}
 B : INT;
 {FLAG OFF}
END_VAR
```

Nenhuma das duas variáveis será inicializada:

```
{FLAG NOINIT ON}
VAR
A : INT;
B : INT;
END_VAR
{FLAG OFF}
VAR
{FLAG NOINIT ON}
A : INT;
B : INT;
{FLAG OFF}
END_VAR
```

Obtendo variáveis para o arquivo de símbolos

Os sinalizadores "**noread**" e "**nowrite**" são usados em uma POU que tenha permissão de leitura e/ou escrita, para fornecer as variáveis selecionadas com direitos de acesso restritos. O padrão para a variável é o mesmo que a configuração para a POU na qual a variável é declarada. Se uma variável não tem permissão de leitura nem de escrita, ela não será exportada para o arquivo de símbolo.

Exemplos:

Se a POU tem permissão de leitura e escrita, então com as seguintes variáveis pragma apenas uma pode ser exportada com permissão de escrita, enquanto a variável b não pode ser exportada:

```
VAR
 A: INT {FLAG NOREAD};
 B: INT {FLAG NOREAD, NOWRITE};
END_VAR
VAR
 {FLAG NOREAD ON}
 A : INT;
 {FLAG NOREAD, NOWRITE ON}
 B : INT;
 {FLAG OFF}
END_VAR
```

Nenhuma das duas variáveis (a/b) será exportada para o arquivo de símbolos:

```
{FLAG NOREAD, NOWRITE ON }
VAR
 A: INT;
 B: INT;
END_VAR
{FLAG OFF}
```

```
VAR
 {FLAG NOREAD, NOWRITE ON}
 A : INT;
 B : INT;
 {FLAG OFF}
END_VAR
```

O pragma adicionalmente opera em todas as declarações de variáveis subseqüentes

Exemplo: todas as POUs em uso serão exportadas com permissão de leitura e escrita.

```
A: AFB;
...
FUNCTION_BLOCK AFB
VAR
 B: BFB {FLAG NOWRITE};
 C: INT;
END_VAR
...
FUNCTION_BLOCK BFB
VAR
 D : INT {FLAG NOREAD};
 E : INT {FLAG NOWRITE};
END_VAR
```

"a.b.d": não será exportado.

"a.b.e": será exportado somente com permissão de leitura.

"a.c": será exportado com permissão de leitura e escrita.

Pragma {bitaccess...} para o acesso a bit

Este pragma pode ser usado para obter uma correta exibição de uma variável que está fazendo um acesso de bit com a ajuda de uma constante global, no assistente de entrada, na função IntelliSense e no monitoramento na janela de declaração. Posteriormente, o efeito será que, quando essa variável for monitorada na janela de declaração da POU específica, as constantes globais usadas serão mostradas abaixo da respectiva variável de estrutura.

Considerar: a opção do projeto 'Substituir constantes' (categoria Compilar) deve ser ativada!

O pragma deve ser inserido na declaração da estrutura em uma linha separada. A linha não é encerrada por um ponto-e-vírgula.

Sintaxe: {bitaccess <Constante Global> <Número do bit> '<comentário>'}

<Constante Global>: nome da constante global que deve ser definido em uma lista de variáveis globais.

<Número do bit >: valor da constante global, conforme definido na lista de variáveis globais.

Para um exemplo, consultar “Endereçando bits nas variáveis”.

Pragma {link} para vincular uma POU durante a geração de código

Normalmente uma POU (programa, função, bloco funcional) ou uma definição de tipo de unidade de dados (DUT) que não é chamada dentro do projeto, não será vinculada durante a geração de código. Mas pode ser desejado que uma função, por exemplo, incluída no projeto por meio de uma biblioteca, esteja disponível após o download do sistema no controlador, mesmo se não for usado diretamente pelo programa aplicativo (por exemplo, para qualquer operação de verificação). Para esta

finalidade, então, pode-se adicionar o pragma {link} em qualquer posição desejada na parte de declaração de uma POU ou em um DUT para forçar uma vinculação da POU.

Pragmas para Controlar a Exibição de Partes de Declaração de Bibliotecas

Durante a criação de uma biblioteca no MasterTool IEC pode-se definir, via pragmas, quais partes da janela de declaração devem ser visíveis e quais não no Gerenciador de Bibliotecas posteriormente, quando a biblioteca for incluída em um projeto. A exibição da parte da implementação da biblioteca não será afetada por isto.

Assim, comentários ou quaisquer declarações de variáveis podem ser escondidos do usuário. Os pragmas {library private} e {library public} afetarão o restante da mesma linha e das linhas subseqüentes, desde que elas não sejam sobrescritas pelas outras.

Sintaxe: {library public}. O teste subseqüente será exibido no Gerenciador de Bibliotecas. {library private}. O teste subseqüente não será exibido.

Exemplo: veja abaixo a parte de declaração de uma biblioteca criada no MasterTool IEC. O comentário "(* this is for all *)" deve ser exibido no Gerenciador de Bibliotecas após ter incluído a biblioteca em um projeto. Entretanto, o comentário "(* but this is not for all *)" não deve ser exibido. As variáveis “local” e “in2” também não devem ser exibidas:

```
{LIBRARY PUBLIC}(* THIS IS FOR ALL *) {LIBRARY PRIVATE} (* THIS IS NOT FOR ALL *)
{LIBRARY PUBLIC}
FUNCTION AFUN : BOOL
VAR_INPUT
IN: BOOL;
END_VAR
{LIBRARY PRIVATE}
VAR
LOCAL: BOOL;
END_VAR
{LIBRARY PUBLIC}
VAR_INPUT
IN2: BOOL;
{LIBRARY PRIVATE}
IN3: BOOL;
{LIBRARY PUBLIC}
END_VAR
```

Instruções Pragma para Entradas do Gerenciador de Parâmetros

Instruções Pragma podem ser inseridas em declarações de variáveis para criar entradas para essas variáveis nas listas de parâmetros (tratadas no Gerenciador de Parâmetros). Dependerá do sistema do dispositivo, se o Gerenciador de Parâmetros está disponível no sistema de programação MasterTool IEC. Isso significa que ele deve ser ativado nas configurações do dispositivo, categoria Funcionalidade de rede.

A sintaxe:

A instrução pragma está entre chaves. Maiúsculas e minúsculas são ignoradas: {<Texto da Instrução>}. Se ele estiver incluído em declarações de variáveis "normais", ele deve ser definido antes do ponto-e-vírgula de terminação da declaração.

Pragmas, usados na interface VAR_CONFIG, são definidos cada um em uma única linha e não são finalizados com um ponto-e-vírgula!

<name>: nome da lista de parâmetros no gerenciador de parâmetros. Se a lista ainda não existir, ela será criada.

<key>: nome do atributo, isto é, o título da coluna na lista de parâmetros, ex.: "Nome", "Valor", "Nível de acesso" etc.; dependerá do tipo de lista de parâmetros específicos do cliente, quais chaves poderão ser definidas no pragma. As definições de chaves são gravadas separadamente por caracteres de espaço, todos juntos entre colchetes. Considerar a sintaxe para entradas nas listas de instância de matrizes, estrutura e componentes de blocos funcionais (veja abaixo, 3).

<value>: valor do atributo que é definido por <key>. Se <value> contém espaços vazios ele deverá ficar entre aspas duplas, por exemplo,... accessright="read only"...

> Nota: as instruções pragma serão efetivas assim que o foco for alterado, isto é, assim que uma pré-compilação for feita, por exemplo, saindo da janela do editor. Entradas erradas não gerarão mensagens até que o projeto seja compilado.

As entradas a seguir podem ser geradas:

1. Entradas na lista de parâmetro de tipo 'Variáveis'

(a) A partir da parte de declaração de programas e listas de variáveis globais

Para uma variável definida em uma declaração de PROGRAMA ou VAR_GLOBAL uma entrada em uma lista de parâmetro do tipo 'Variáveis' pode ser criada, se ela estiver declarada como descrito a seguir (se a lista de parâmetros ainda não existir, ela será criada automaticamente):

Sintaxe: {parameter list=<name> [<key>=<value> <key>=<value> ...mais teclas] }

Exemplo: variável bvar é declarada em um programa. Ela deve ser inserida na lista de parâmetros parlist1 (tipo 'Variáveis') com o nome simbólico bvar1, valor 102, índice 16#1200 e sub-índice 16#2l.

```
VAR
BVAR:INT{PARAMETER LIST=PARLIST1 [NAME=BVAR1 VALUE=102 INDEX=16#1200 SUBINDEX=16#1 ] };
END_VAR
```

(b) Por meio de uma declaração na interface VAR_CONFIG

Há a possibilidade de criar uma entrada para uma variável em uma lista de parâmetros de tipo 'Variáveis' colocando um pragma em uma janela VAR_CONFIG (independente das configurações da variável, que também são feitas na interface do VAR_CONFIG!). Se a lista de parâmetros ainda não existir, ela será criada automaticamente.

Sintaxe: {parameter list=<name> path=<path> [<key>=<value> <key>=<value> ...mais teclas] }

<path> o caminho da variável para o qual a entrada deve ser gerada, por exemplo, "PLC_PRG.act1.var_x"

Exemplo: para a variável var_x uma entrada é criada na lista de parâmetros "varlist1", o nome simbólico é "xvar".

```
VAR_CONFIG
{PARAMETER LIST=VARLIST1 PATH=PLC_PRG.ACT1.VAR_X [ NAME=XVAR ]
END_VAR
```

2 Entradas em listas de parâmetros de tipo 'Modelo' por meio de blocos funcionais e estruturas

Podem ser usados pragmas em declarações de variáveis em blocos funcionais e estruturas para criar entradas nas listas de parâmetro de tipo 'Modelo'. Se o modelo ainda não existir, ele será criado automaticamente.

Sintaxe: {template list=<name> [<key>=<value> <key>=<value> ...mais teclas…] }:

Exemplo: variável strvar, que é um elemento da estrutura "stru1", deve ser inserida em uma lista de parâmetros "templ1" de tipo 'Modelo'; o nome simbólico (membro) da entrada é "struvar1" e o nível de acesso é "low":

```
TYPE STRU :
  STRUCT
  IVAR:INT;
  STRVAR:STRING{TEMPLATE LIST=VORL1 [MEMBER=STRUVAR1 ACCESSLEVEL=LOW] };
END_STRUCT
END_TYPE
```

3. Entradas na lista de parâmetros de tipo 'Instância'

(a) Por meio de declarações em programas ou listas de variáveis globais

Na declaração de matrizes, bloco funcional - ou estrutura de variáveis em um programa ou em uma lista de variáveis globais, uma lista de parâmetro de tipo 'Instância' pode ser criada diretamente:

Sintaxe: {instance list=<name> template=<template> baseindex=<index> basesubindex=<subindex> [<key>=<valor para o primeiro elemento> <key>=< valor para o primeiro elemento > ...mais teclas para o primeiro elemento] | [<key>=< valor para o segundo elemento > <key>=< valor para o segundo elemento > .. mais teclas para o segundo elemento] | [teclas para elementos adicionais]}<key>=<value> <key>=<value> ...mais teclas] }

Para matrizes, a chave "template" será automaticamente definida com o modelo "ARRAY" disponível implicitamente. Para estruturas e blocos funcionais deve estar disponível no Gerenciador de parâmetros um modelo apropriado e também deve ser parte da declaração descrita aqui.

Para cada matriz específica e para cada elemento de estrutura ou bloco funcional, uma entrada individual pode ser predefinida na lista de parâmetros, por exemplo, para cada elemento uma definição própria [name=<nomedoelemento>] pode ser especificada.

As definições de chaves de cada elemento específico (todas as chaves de um elemento específico devem ser colocadas pelos mesmos colchetes!) são organizados em uma linha, separada por espaços vazios. Os pacotes de definição para os elementos automaticamente se referirão aos elementos de acordo com a ordem crescente de índice (Member). Se não há tantas chaves consultar o exemplo 1b para referência.

Opções relacionadas à chave "Nome" ao inserir matrizes nas listas de parâmetro

Se não for especificado um nome para um elemento de matriz no pragma, esse elemento e todos os elementos seguintes automaticamente obterão os nomes <nome da POU>_<nome da variável matriz>_<número de índice da matriz correspondente>.

Exemplo: matriz variável ARRVAR [1..8] de INT na POU PLC_PRG deve ser inserida em uma lista de variáveis via um pragma; se a chave "name" não está definida no pragma, os elementos de matriz específicos na lista de parâmetros automaticamente serão nomeados "PLC_PRG_arrvar_1" para "PLC_PRG_arrvar_8".

"<name>_<primeiro número de índice para intervalo de matriz>" para o primeiro elemento da matriz via o pragma, os seguintes elementos de matriz na lista de parâmetros automaticamente serão nomeados “<name>_<número de índice correspondente>”.

Exemplo: matriz variável ARRVAR [1..8] de INT a ser inserida em uma lista de variáveis via um pragma; O pragma especifica apenas [name=xyz_1] para o primeiro elemento da matriz e, portanto, na lista de parâmetros, os elementos seguintes automaticamente irão obter os nomes "xyz_2" a "xyz_8".

Atenção: não definir um valor para a chave "Member"; esta coluna será preenchida automaticamente usando os valores de índice de matriz.

Exemplos:

Exemplo1a: uma variável matriz arr_1 é declarada como descrito nas seguintes opções para criar uma Lista de Parâmetros arrinst de tipo 'Instance' (se não ainda disponível), na qual os componentes da matriz são inseridos (cada elemento primeiro obterá um nome simbólico xname_<número de índice>

(pode ser editado posteriormente no Gerenciador de parâmetros) e o sub-índice irá ser incrementado por 1 para cada entrada, começando com 0 (basesubindex). Accesslevel=LOW será aplicada a todos os elementos.

arr_1: ARRAY [1..8] OF INT{instance list=arrinst template=ARRAY baseindex=16#0 basesubindex=16#0 [name=xname_1 accesslevel=low] };

Exemplo1b: para uma variável matriz arr_1 apenas para os elementos de 1 a 4 nomes já estão definidos por meio do pragma; os elementos de 5 a 8, portanto, automaticamente irão obter o nome do elemento 4 estendido por um sublinhado e o número de índice apropriado, isto é, "xname_5" até "xname_8". Considere que as definições de chave adicional para um elemento específico deve ser inserido dentro dos mesmos colchetes conforme mostrado aqui para o primeiro e o quarto elemento em relação à accesslevel:

arr_1: ARRAY [1..8] OF INT{instance list=arrinst template=ARRAY baseindex=16#0 basesubindex=16#0 [name=aname accesslevel=high][name=bname][name=cname][name=xname accesslevel=medium]};

Editor do Gerenciador de Parâmetros, Exemplo1a:

arrinst

Name	Member	Val...	Index	SubIn...	Accesslevel	Acce...	Min	Max
xname_1	[1]		16#0	16#0	low			
xname_2	[2]		16#0	16#1	low			
xname_3	[3]		16#0	16#2	low			
xname_4	[4]		16#0	16#3	low			
xname_5	[5]		16#0	16#4	low			
xname_6	[6]		16#0	16#5	low			
xname_7	[7]		16#0	16#6	low			
xname_8	[8]		16#0	16#7	low			

Synchronous actions ☑ Template ARRAY Base index 16#0
Base variable PLC_PRG.arr_1 Base subindex 16#0
Apply

Figura 5-8. Editor do Gerenciador de Parâmetros, Exemplo1a

Editor do Gerenciador de Parâmetros, Exemplo1b:

arrinst

Name	Member	Value	Index	SubIn...	Accesslevel	Acces...	Min	Max
aname	[1]		16#0	16#0	high			
bname	[2]		16#0	16#1	medium			
cname	[3]		16#0	16#2	medium			
xname	[4]		16#0	16#3	medium			
xname_5	[5]		16#0	16#4	medium			
xname_6	[6]		16#0	16#5	medium			
xname_7	[7]		16#0	16#6	medium			
xname_8	[8]		16#0	16#7	medium			

Synchronous actions ☑ Template ARRAY Base index 16#0
Base variable PLC_PRG.arr_1 Base subindex 16#0
Apply

Figura 5-9. Editor do Gerenciador de Parâmetros, Exemplo1b

Exemplo 2:

Uma variável de estrutura de tipo stru1 (contém variáveis a, b, c) está declarada como descrito nas seguintes opções para obter entradas em uma lista de parâmetros do tipo 'Instância' que é baseado no modelo strulist_temp; a lista obterá as entradas para os componentes a, b, c, nomes simbólicos não são atribuídos, o nível de acesso é definido como "high" e cada valor de índice definido pelo modelo

será aumentado por 2. Verifique se o modelo definido no pragma está disponível no Gerenciador de parâmetros:

struvar:stru1{instance list=strulist template=strulist_templ baseindex=16#2 basesubindex=16#0 [accesslevel=high] };

Exemplo 2, entradas para a variável de estrutura no modelo:

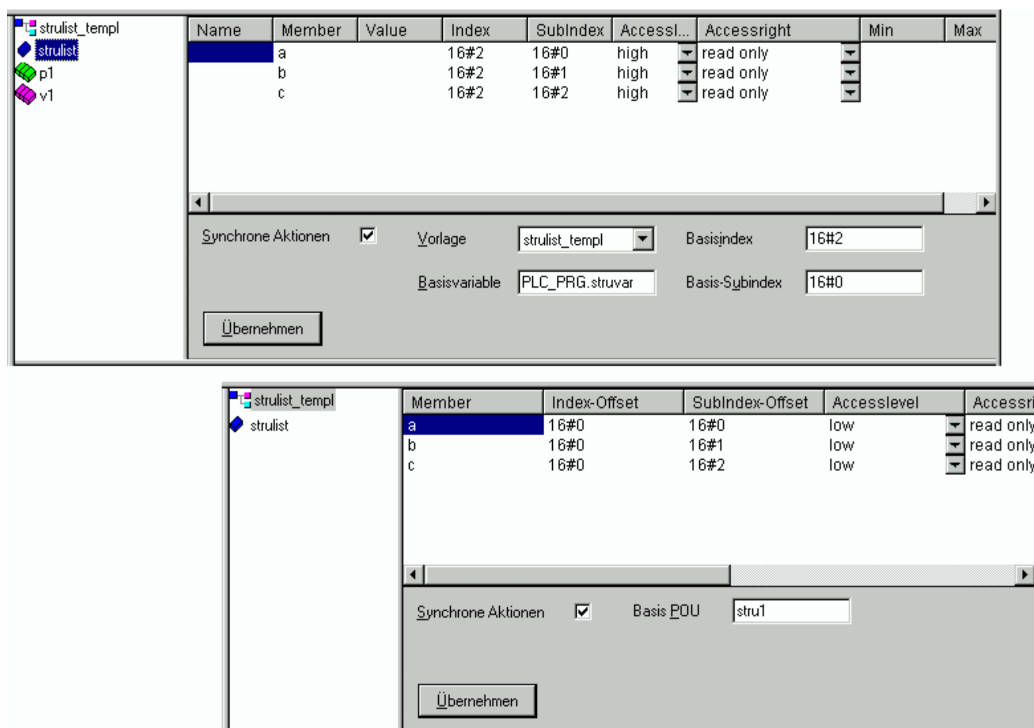


Figura 5-10. Exemplo 2, entradas para a variável de estrutura no modelo

(b) Por meio de declarações da interface VAR_CONFIG

Para variáveis que podem ser instanciadas, pode-se diretamente criar entradas em uma lista de parâmetros do tipo 'Instância' por um pragma em uma janela VAR_CONFIG (independentemente de definições de configuração de variável que também são feitas nesta interface!).

Nota: se o modelo ainda não existir, ele será criado automaticamente.

Verifique se o modelo definido no pragma já está disponível no Gerenciador de parâmetro.

Sintaxe: {instance list=<name> path=<path> template=<template> baseindex=<index> basesubindex=<subindex>[<key>=<value> <key>=<value> ...mais teclas] }

<path>: o caminho da instância da variável; ex. "PLC_PRG.fb1inst", (fb1inst é uma instância do bloco funcional FB1).

Exemplo: a seguinte entrada em uma janela VAR_CONFIG criará entradas para todas as variáveis do bloco funcional FB1 em uma lista de instância varinst1 baseada no modelo fb1_templ (que deve já estar disponível). Para cada entrada o offset do índice, que é predefinido pelo modelo, será incrementado por 2 (baseindex), o offset do sub-índice não será modificado (basesubindex). Cada entrada obtém um nome simbólico fb1var, que deve-se editar posteriormente no Gerenciador de parâmetros.

```
VAR_CONFIG

{INSTANCE LIST=VARINST1 PATH=PLC_PRG.FB1 TEMPLATE=FB1_TEMPL BASEINDEX=16#2
BASESUBINDEX=16#0 [ NAME=FB1VAR ]}

END_VAR
```

Pragma para tipos de dados não persistentes

Normalmente o seguinte é válido: mesmo se somente uma variável local em um bloco funcional ou uma estrutura é declarada persistente, no uso de uma instância, automaticamente todos os componentes serão armazenados nas informações persistentes (persist.dat) no controlador. Para economizar espaço, pode-se usar o pragma

{nonpersistent}

na declaração do bloco funcional e na estrutura. O efeito disto é que somente aqueles componentes da estrutura e do bloco funcional, que são declaradas como "persistentes", serão inseridos nas informações persistentes.

Exemplo:

Se uma instância do seguinte bloco funcional é declarada como persistente, somente as variáveis locais e fblevel3 serão gravados para as informações persistentes. Sem o pragma {nonpersistent} todas as variáveis do bloco funcional seriam armazenadas lá.

```
FUNCTION_BLOCK FB_LEVEL_2
{NONPERSISTENT}
VAR_INPUT
BVAR_IN : BOOL;
END_VAR
VAR_OUTPUT
BVAR_OUT : BOOL;
END_VAR
VAR
IVAR2 : INT;
END_VAR
VAR PERSISTENT
LOCAL : INT := 33;
FBLEVEL3 : FB_LEVEL_3;
END_VAR
```

Editores de texto

Trabalhando em editores de texto

Os editores de texto usados para a parte de implementação (Editor IL / ST) do MasterTool IEC fornecem as funções usuais de editor de texto do Windows.

A implementação nos editores de texto é suportada pela cor de sintaxe.

No modo de sobreposição a barra de status mostra um **OV** preto. Pode-se alternar entre o modo de sobreposição e modo de inserção pela tecla <Ins>.

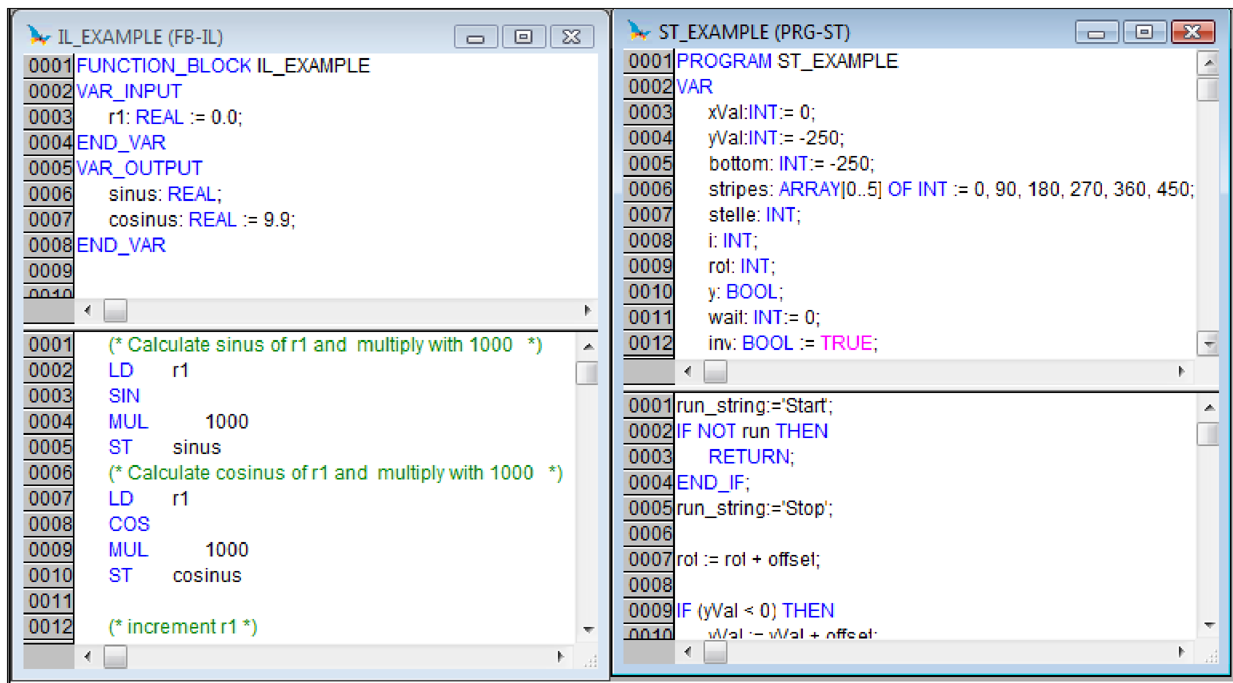


Figura 5-11. Implementação nos editores de texto

Os comandos mais importantes são encontrados no menu de contexto (botão direito do mouse ou <Ctrl> + <F10>).

Os editores de texto usam os seguintes comandos de menu...

'Inserir' 'Operadores' nos editores de texto

Com este comando todos os operadores disponíveis na linguagem atual são exibidos em uma caixa de diálogo.

Se um dos operadores é selecionado e a lista for fechada com **OK**, então, o operador realçado será inserido na posição atual do cursor (isso é gerenciado da mesma forma no Assistente de Entrada).

'Inserir' 'Operandos' nos editores de texto

Com este comando todas as variáveis em uma caixa de diálogo são exibidas. Pode-se selecionar se deseja exibir uma lista das variáveis globais, locais ou as do sistema.

Se um dos operandos for escolhido, e a caixa de diálogo é fechada com **OK**, o operando realçado será inserido na posição atual do cursor (isso é gerenciado da mesma forma no Assistente de Entrada).

'Inserir' 'Função' nos editores de texto

Com este comando todas as funções serão exibidas em uma caixa de diálogo. Pode-se escolher se quer a exibição da lista de funções definida pelo usuário ou padrão.

Se uma das funções é selecionada e a caixa de diálogo é fechada com **OK**, então, a função realçada será inserida na posição atual do cursor (o gerenciamento irá continuar, como na seleção de entrada).

Se a opção **Com argumentos** foi selecionada na caixa de diálogo, então, as variáveis de entrada e saída necessárias também serão inseridas.

'Inserir' ' Bloco funcional ' nos editores de texto

Com este comando todos os blocos funcionais são exibidos em uma caixa de diálogo. Pode-se optar por ter uma lista exibindo blocos funcionais definida pelo usuário ou padrão.

Se um dos blocos funcionais está selecionado e a caixa de diálogo é fechada com **OK**, então, o bloco funcional realçado será inserido na posição atual do cursor (isso é gerenciado da mesma forma no Assistente de Entrada).

Se a opção **Com argumentos** foi selecionada na caixa de diálogo, as variáveis de entrada do bloco funcional necessárias também serão inseridas. No entanto não é obrigatória a atribuição desses parâmetros.

Chamando POUs com parâmetros de saída nos editores de texto

Os parâmetros de saída de uma POU chamada podem ser atribuídos diretamente após a chamada nas linguagens textuais IL e ST.

Exemplo:

Parâmetro de saída out1 de afbinst é atribuído a variável a.

IL: `CAL AFBINST(IN1:=1, OUT1=>A)`

ST: `AFBINST(IN1:=1, OUT1=>A);`

Se a POU é inserida por meio do Assistente de Entrada (<F2>) com a opção 'Com argumentos' na janela de implementação de uma POU ST ou IL, ela automaticamente será exibida com todos os parâmetros dessa sintaxe. No entanto não é obrigatória a atribuição desses parâmetros.

Os editores de texto no modo online

As funções online nos editores são configuradas no processamento (etapas) do breakpoint e ciclado. Junto com a monitoração, o usuário possui a capacidade de depuração a partir de um moderno depurador de linguagem padrão do Windows.

No modo online, a janela do editor de texto é dividida verticalmente em metades. No lado esquerdo da janela encontra-se o programa de texto normal; no lado direito, aparecerá uma exibição das variáveis cujos valores foram alterados nas respectivas linhas. A largura das metades da janela pode ser modificada desenhando a linha divisória com o mouse.

A exibição é a mesma que na parte de declarações. O que significa que quando o CP está em execução, os valores presentes nas respectivas variáveis serão exibidos.

O seguinte deve ser observado durante a monitoração de expressões ou variáveis endereçadas de bit: no caso de expressões, o valor da expressão inteira sempre é exibido. Exemplo: a AND b são exibidas em azul ou com ":=TRUE" se ambos, a e b são TRUE. Para variáveis endereçadas de bit, o valor do bit que é endereçado sempre é monitorado (ex.: a.3 é exibido em azul ou com ":=TRUE" se a tem o valor apropriado).

Ao colocar o ponteiro do mouse brevemente acima de uma variável, o tipo, o endereço e o comentário sobre a variável serão exibidos em uma Tooltip.

'Extras' 'Opções de monitoração'

Com esse comando pode-se configurar sua janela de monitoração. Nos editores de texto, a janela é dividida em duas metades durante a monitoração. O programa está localizado na metade esquerda. Na metade direita, todas as variáveis que estão localizadas na linha correspondente do programa são monitoradas.

Pode-se especificar a **largura** da janela do monitor e qual **distância** duas variáveis devem ter em uma linha. Uma declaração de distância de 1 corresponde, nesse caso, a uma altura de linha na fonte selecionada.

Considere que a largura das metades janela também pode ser modificada desenhando a linha divisória com o mouse.

Caixa de diálogo Opções de Monitoração:

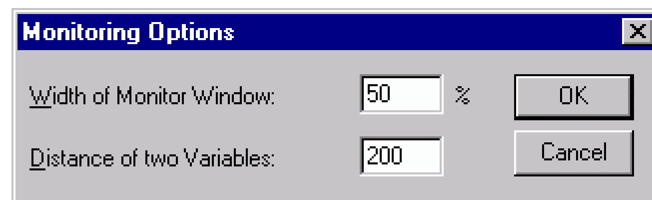


Figura 5-12. Caixa de diálogo Opções de Monitoração

Posições do Breakpoint no editor de texto

Posições de breakpoint incluem todas as posições em um programa nas quais valores de variáveis podem sofrer alteração ou onde o fluxo do programa se ramifica. Exceção: chamadas de função (se necessário, um breakpoint na função deve ser definido aqui). Nas posições intermediárias, um breakpoint não faria sentido, desde que nada é capaz de alterar nos dados desde a posição do breakpoint anterior.

Isso resulta nas seguintes posições de breakpoint no IL:

- No início da POU
- Em cada LD, LDN (ou, no caso do LD estar localizado em um rótulo, então, no rótulo)
- Em cada JMPCN JMP, JMPC,
- Em cada rótulo
- Em cada CALCN CAL, CALC,
- Em cada RETCN RET, RETC,
- No final da POU

Texto Estruturado acomoda as seguintes posições de breakpoint:

- Em cada atribuição
- Em cada instrução RETURN e EXIT
- Nas linhas em que as condições estão sendo avaliadas (WHILE, IF, REPEAT)
- No final da POU

Posições de Breakpoint serão marcadas com a exibição do campo de número de linhas na cor definida nas opções do projeto.

Editor IL com possíveis posições de breakpoint (campos de número):

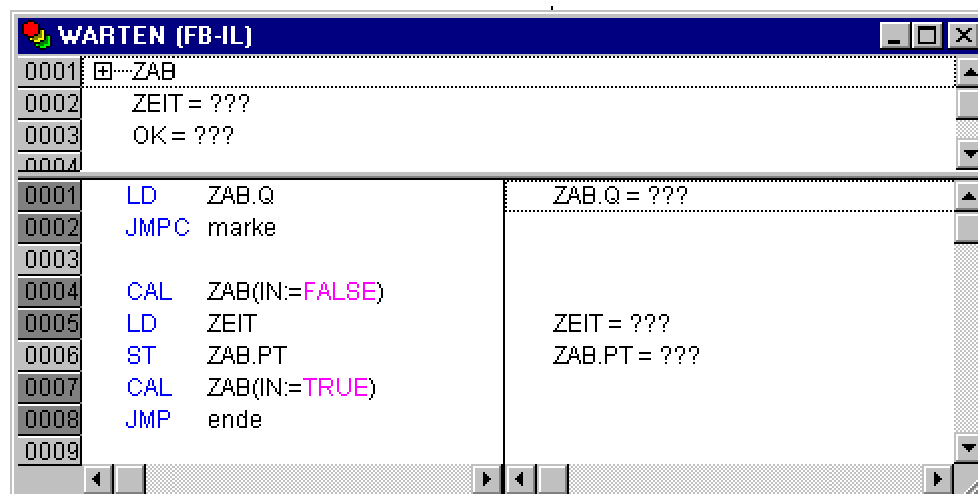


Figura 5-13. Editor IL com possíveis posições de breakpoint

Como configurar um breakpoint?

Para definir um breakpoint, clique no campo do número da linha na linha onde se deseja definir um breakpoint. Se o campo selecionado for uma posição do breakpoint, a cor do campo de números de linha será alterada de cinza escura para azul claro e o breakpoint será ativado no CP.

Excluindo breakpoints

Para excluir um breakpoint, clique no campo do número de linha da linha com o breakpoint a ser excluído.

A configuração e exclusão de breakpoints podem também ser selecionados através do menu (‘Comunicação’ 'Alternar breakpoint'), via tecla de função <F9> ou através do símbolo na barra de ferramentas.

O que acontece em um breakpoint?

Se um breakpoint é alcançado no CP, então a tela exibirá a quebra com a linha correspondente. O campo do número da linha na linha onde o CP está posicionado, será exibido em vermelho. O programa de usuário é interrompido no CP.

Se o programa estiver em um breakpoint, então o processamento pode ser reiniciado com ‘Comunicação’ 'RUN'.

Além disso, com ‘Comunicação’ 'Sobrepasso' ou 'Passo a passo' pode-se fazer o programa ser executado na próxima posição do breakpoint. Se a instrução for um comando de CAL, ou, se houver uma chamada de função nas linhas acima da próxima posição do breakpoint, então, poder-se-á usar 'Sobrepasso' para ignorar a chamada de função. Com 'Passo a passo', se ramificará a POU aberta.

Número da linha do editor de texto

Os números de linha do editor de texto fornecem o número de cada linha de texto de uma implementação de uma POU.

No modo offline, um clique simples em um número de linha especial marcará a linha inteira de texto.

No modo online, a cor do plano de fundo do número da linha indica o status de breakpoint de cada linha. As configurações padrão para as cores são:

- Cinza escuro: essa linha é uma posição possível para um breakpoint
- Azul claro: um breakpoint foi definido nessa linha
- Vermelha: o programa atingiu esse ponto

No modo online, simplesmente clicando com o mouse irá alterar o status do breakpoint desta linha.

O editor IL

Esta é a aparência de uma POU escrita em IL sob o editor correspondente do MasterTool IEC:

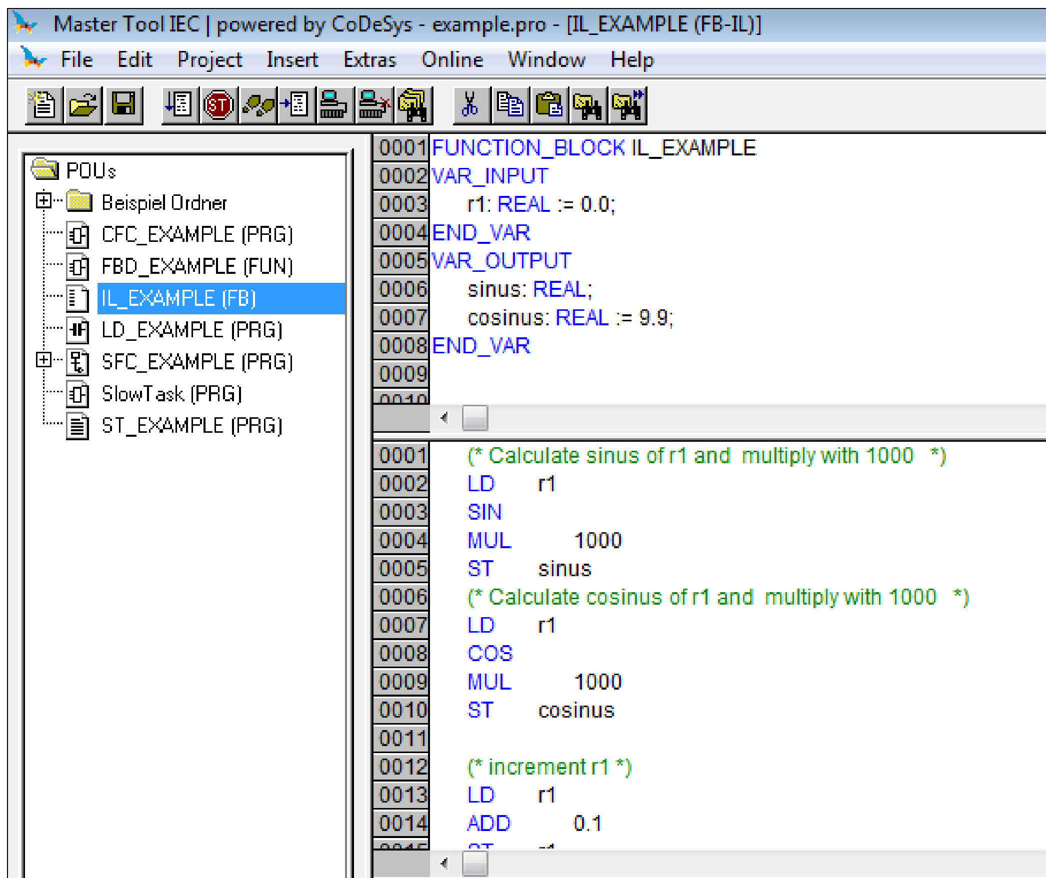


Figura 5-14. Aparência de uma POU gravada em IL

Todos os editores para POUs consistem de uma parte de declaração e de um corpo. Estas partes são separadas por um divisor de tela.

O editor de Lista de Instrução é um editor de texto com as capacidades habituais de editores de texto do Windows. Os comandos mais importante são encontrados no menu de contexto (botão direito do mouse ou <Ctrl> + <F10>). Chamadas múltiplas de POU também são possíveis

Exemplo:

```
CAL CTU_INST(
CU:=%IX10,
PV:=(
LD A
ADD 5
)
)
```

Para obter informações sobre a linguagem, consultar o capítulo 2, Lista de Instruções (IL).

IL no modo online

Com o comando ‘Comunicação’ 'Exibir controle de fluxo', um campo adicional na qual o conteúdo do acumulador é exibido é inserido no editor de IL no lado esquerdo de cada linha.

Para mais informações sobre o editor de IL no modo Online, consultar 'Os Editores de Texto no Modo Online'.

O editor ST

Veja a seguir como uma POU escrita em ST aparece sob o editor correspondente do MasterTool IEC:

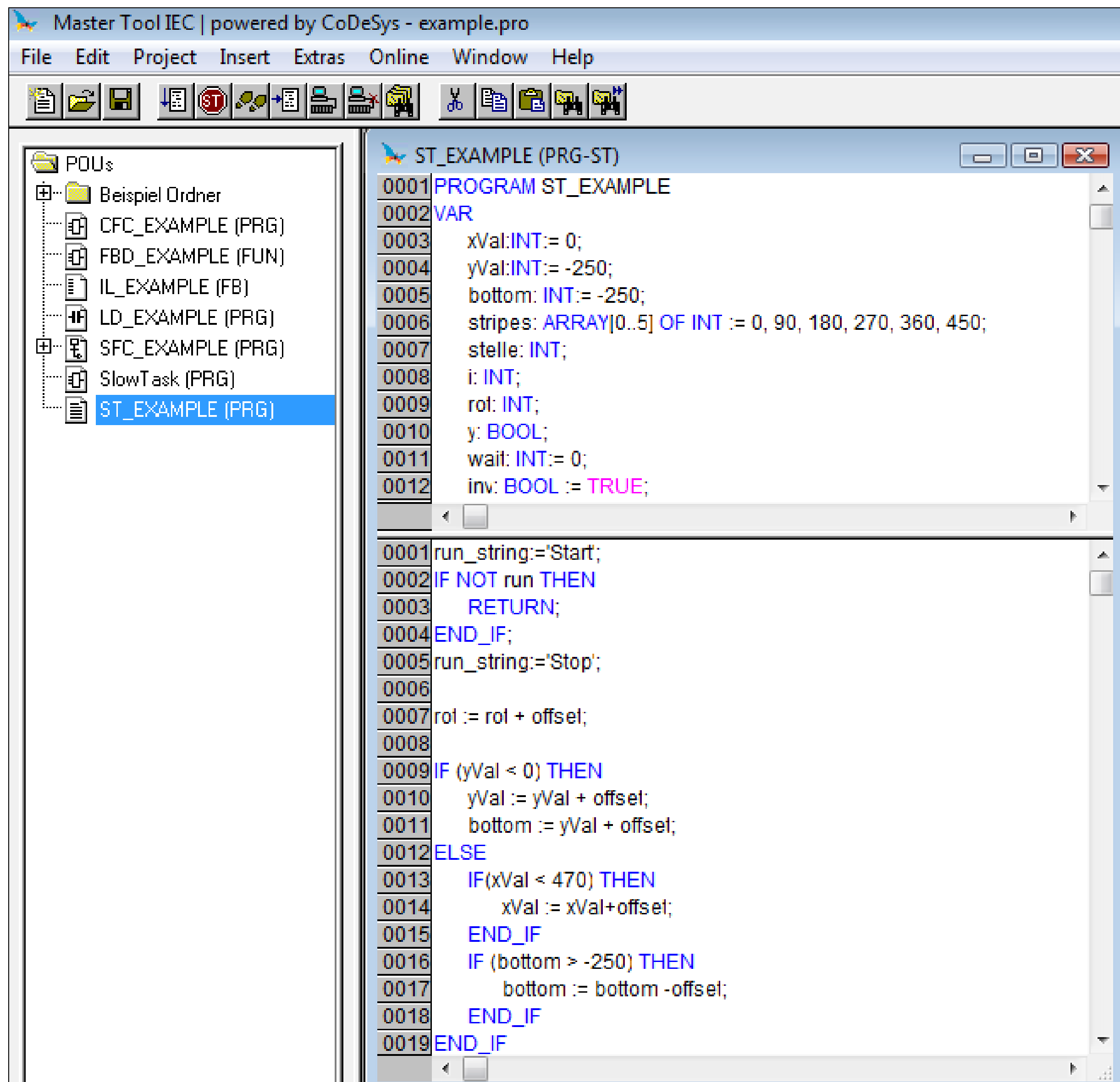


Figura 5-15. POU escrita em ST

Todos os editores para POUs consistem de uma parte de declaração e um corpo. Estas partes são separadas por um divisor de tela.

O editor de texto estruturado é um editor de texto com as capacidades habituais de editores de texto do Windows. Os comandos mais importantes são encontrados no menu de contexto (botão direito do mouse ou <Ctrl> + <F10>).

Para obter informações sobre a linguagem, consultar o capítulo 2, Texto Estruturado (ST).

Os editores gráficos

Trabalhando em editores gráficos

Os editores de linguagem graficamente orientados, SFC (Seqüenciamento Gráfico de Funções), LD (Diagrama Ladder) e FBD (Diagrama de Blocos Funcionais) e os diagramas de blocos funcionais gráficos livres têm muitos pontos em comum.

Nos parágrafos a seguir esses recursos serão sumarizados; as descrições específicas do LD, FBD e CFC, bem como a linguagem SFC, seguem em seções separadas. A implementação nos editores gráficos é suportada pela cor da sintaxe.

Zoom

Objetos, como POUs, ações, transições, etc. nas linguagens SFC, LD, FBD, CFC e em visualizações podem ser ampliados ou reduzidos no tamanho com uma função de zoom. Todos os elementos do conteúdo da janela da parte de implementação são afetados; a parte da declaração permanece inalterada.

No formato padrão, cada objeto é exibido com nível de zoom de100 %. O nível de zoom que é definido será salvo como uma propriedade de objeto no projeto.

A impressão de documentação do projeto sempre ocorre no nível de exibição de 100 %!

O nível de zoom pode ser definido por uma lista de seleção na barra de ferramentas. Os valores entre 25 % e 400 % podem ser selecionados; valores individuais entre 10 % e 500% podem ser inseridos manualmente.

A seleção de um nível de zoom só estará disponível se o cursor estiver posicionado em um objeto criado em uma linguagem gráfica ou em um objeto de visualização.

Mesmo com o objeto submetido ao Zoom, as posições do cursor nos editores podem ser selecionadas e alcançadas com as teclas de direção. O tamanho do texto é controlado pelo fator de zoom e o tamanho da fonte que está definido.

A execução de todos os recursos do menu editor (ex.: inserir uma caixa) como uma função da posição do cursor está disponível em todos os níveis de zoom, tomando o mesmo em consideração.

No modo online, cada objeto é exibido de acordo com o nível de zoom que tenha sido definido; a funcionalidade online está disponível sem restrição.

Quando o IntelliMouse é usado, um objeto pode ser ampliado/reduzido, pressionando a tecla <CTRL> e ao mesmo tempo ativando o scroll do mouse para frente ou para trás.

Rede

Nos editores LD e FBD, o programa é organizado em uma lista de redes. Cada rede é designada no lado esquerdo por um número de rede serial e tem uma estrutura que consiste em uma expressão lógica ou aritmética, um programa, função ou chamada de bloco funcional, e um salto ou uma instrução de retorno.

Rótulo

Cada rede tem um rótulo que pode, opcionalmente, ser deixado vazio. Esse rótulo é editado, clicando na primeira linha da rede, logo a seguir do número da mesma. Agora pode-se inserir um rótulo, seguido de dois pontos.

Comentários, Redes com Quebras de Linha 'Extras' 'Opções'

Cada rede pode ter um comentário multi-linha. Na caixa de diálogo 'Opções de Bloco Funcional e Diagrama Ladder', a qual pode ser aberta, executando o comando 'Extras' 'Opções', pode-se realizar as configurações relacionadas a comentários e quebras de linha.

Opções de Blocos Funcionais e Diagrama LD:

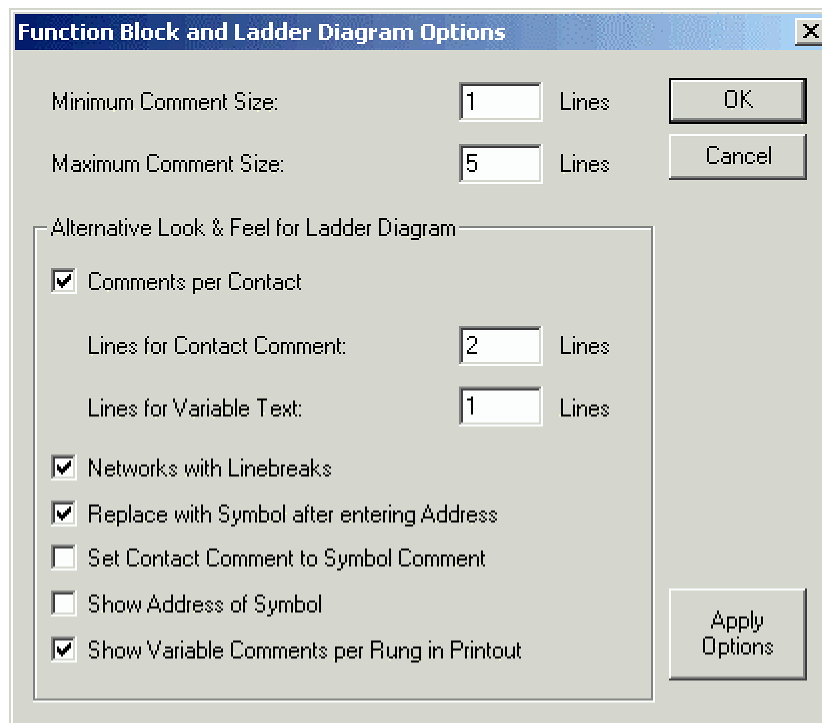


Figura 5-16. Opções de Blocos Funcionais e Diagrama Ladder

Tamanho máximo do comentário: número máximo de linhas a ser disponibilizado para um comentário de rede (o valor padrão aqui é 4.)

Tamanho mínimo do comentário: o número de linhas que normalmente deve ser reservado para comentários. Se, por exemplo, o número 2 é inserido, então, no início de cada rede haverá duas linhas em branco após a linha de rótulo. O valor padrão aqui é 0, o que tem a vantagem de permitir que mais redes caibam na área de tela.Se o tamanho mínimo do comentário for maior que 0, então, para inserir um comentário basta clicar na linha de comentário e, em seguida, inserir o comentário. Caso contrário, primeiro deve ser selecionada a rede na qual um comentário deve ser inserido e então usa-se **'Inserir' 'Comentário'** para inserir uma linha de comentário. Os comentários são exibidos em cinza.

Aparência alternativa: as seguintes opções permitem definir uma aparência alternativa das redes...

Comentários por contato (somente para Editor LD): se esta opção é ativada, pode-se atribuir um comentário individual para cada contato ou bobina. No campo de edição **Linhas para comentário de variáveis** insira o número de linhas que devem ser reservadas e exibidas para o comentário. Se esta configuração for feita, um campo de comentário será exibido no editor acima de cada contato e bobina onde pode-se inserir texto.Se Comentários por contato for ativado, então, no Editor LD também o número de linhas (**linhas de texto variável**:) que deve ser usado para o nome da variável de bobina e contato podem ser definidas e. Isso é usado para exibir nomes longos dividindo-os em várias linhas. No exemplo a seguir 2 linhas são definidas para o comentário de variável e 1 linha para o texto variável:

Exemplo de uma rede com exibição de variáveis, comentários e endereços:

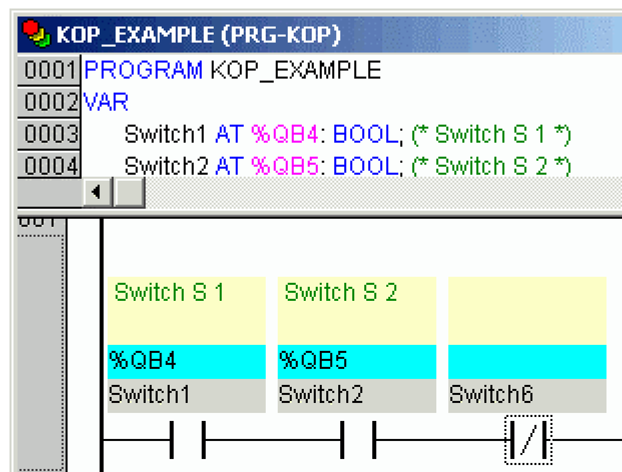


Figura 5-17. Exemplo de uma rede com exibição de variáveis, comentários e endereços

Redes com quebras de linha (somente para o editor LD): se esta opção é ativada, quebras de linha serão forçadas nas redes assim que o comprimento de rede exceder o tamanho de janela determinado (alguns dos elementos não seriam visíveis).

Exemplo de uma rede com quebras de linha:

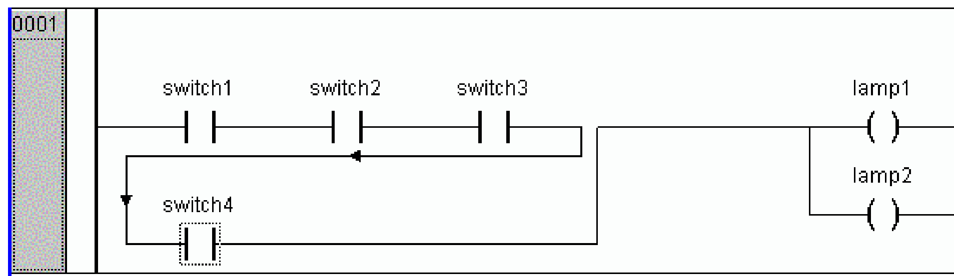


Figura 5-18. Exemplo de uma rede com quebras de linha

Substituir com o símbolo após a inserção do endereço (apenas para Editor LD): se esta opção é ativada, pode-se inserir um endereço em uma Caixa e em um contato ou bobina (por exemplo, "%QB4") e esse endereço será substituído imediatamente pelo nome da variável que é atribuído o endereço. Se não houver nenhuma variável atribuída ao endereço digitado, o endereço permanecerá.

Definir comentário do contato para comentário do símbolo: se esta opção é ativada, no campo comentário de um contato e uma bobina, será exibido o comentário que foi definido na declaração da variável usada para o contato ou bobina. O comentário, então, pode ser editado. (veja exemplo na figura subseqüente.). Para essa finalidade, porém a opção 'Comentários por contato' também deve ser ativada. Considere que um comentário que foi inserido já localmente em um contato ou bobina será substituído automaticamente por comentário variável em qualquer caso, mesmo se a variável não tem um comentário na sua declaração!

Mostrar endereço do símbolo (apenas para Editor Ladder): se esta opção é ativada e uma variável atribuída a um contato ou bobina é atribuída a um endereço, o endereço será exibido acima do nome de variável (veja exemplo na imagem acima).

Mostrar comentários de variável por rede na impressão: se esta opção é ativada, em cada rede, para cada variável usada naquela rede, será exibida uma linha, mostrando o nome, endereço, tipo de dados e comentário para essa variável, conforme definido na declaração de variáveis. Isto pode ser útil para uma documentação (impressão) do projeto.

Exemplo de uma rede mostrando uma linha de informação para cada variável:

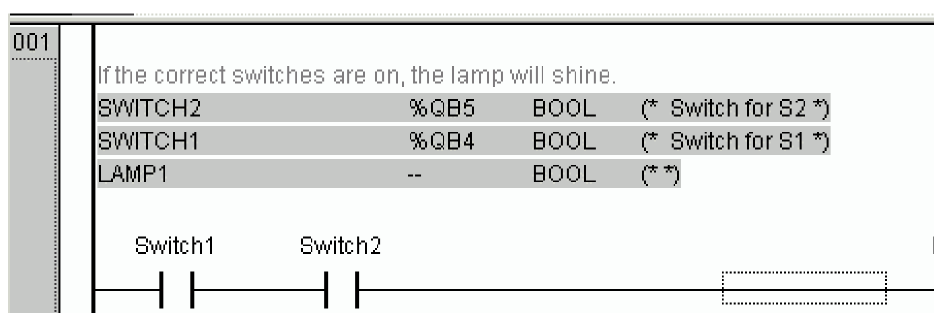


Figura 5-19. Exemplo de uma rede mostrando uma linha de informação para cada variável

Aplicando as opções:

OK: pressione este botão para aplicar as configurações na POU atual e para fechar a caixa de diálogo de opções.

Aplicar opções: pressione este botão para aplicar as configurações do projeto inteiro. Uma caixa de diálogo será aberta para confirmação.

'Inserir''Rede (depois)' ou 'Inserir' 'Rede (antes)'

Atalho: <Shift> + <T> (rede depois)

Para inserir uma nova rede no FBD ou LD, selecione **'Inserir' 'Rede (depois)'** ou o comando **'Inserir' 'Rede (antes)'**, dependendo se deseja-se inserir a nova rede depois ou antes a rede atual. A rede atual pode ser alterada clicando no número de rede. Ela é marcada via retângulo pontilhado sob o número. Com a tecla <Shift> e um clique no mouse é possível selecionar toda a área das redes, a partir da atual para aquela selecionada.

Os editores de rede no modo online

No Editores FBD e LD somente pode-se definir breakpoints para redes. O campo do número da rede de uma rede para o qual um breakpoint foi definido será exibido em azul. O processamento então pára na frente da rede, onde o breakpoint está localizado. Nesse caso, o campo do número de rede é exibido em vermelho. Com processamento ciclado (etapas), pode-se ir de rede para rede.

Todos os valores são monitorados na entrada e saída de redes das POUs (Unidades de Organização de Programa).

O seguinte deve ser observado durante o monitoramento de expressões ou variáveis endereçadas via bit: em expressões, por exemplo, a AND b, usadas como condição de transição ou entrada de blocos funcionais, o valor da expressão inteira sempre é exibida (a AND b são mostradas em azul ou como: =TRUE, se a e b são TRUE). Para variáveis endereçadas via bit, o valor de bits que é endereçado sempre é monitorado (ex.: a.3 é exibido em azul ou com „:=TRUE, se a tem o valor 4)

O controle de fluxo é executado com o comando ‘Comunicação’ 'Exibir controle de fluxo' . Usando o controle de fluxo, pode-se exibir os valores presentes que estão sendo executados nas redes através de linhas de conexão. Se as linhas de conexão não “transmitem” valores booleanos, o valor será exibido em um campo especialmente inserido. Os campos de monitor para variáveis que não são usadas (por exemplo, na função SEL) são exibidos em um tom de cinza. Se as linhas “transmitem” valores booleanos, então, elas serão sombreadas em azul, no caso de carregar TRUE. Portanto, pode-se acompanhar o fluxo de informações enquanto o CP está em execução.

Ao colocar o ponteiro do mouse rapidamente acima de uma variável, então, o tipo, o endereço e o comentário da variável serão exibidos em uma Tooltip.

O editor FBD

Veja a seguir a aparência de uma POU escrita em FBD sob o editor MasterTool IEC correspondente:

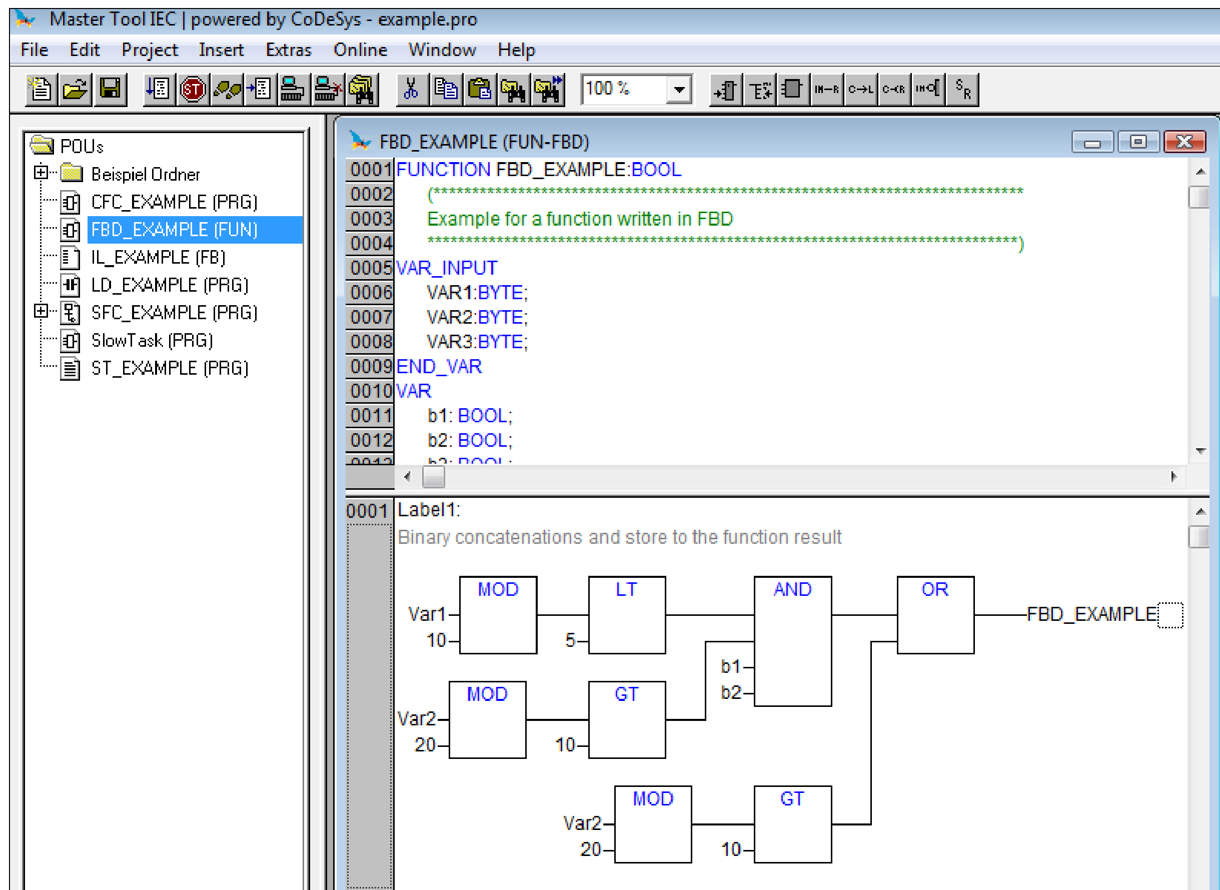


Figura 5-20. Editor FBD

O Editor FBD é um editor gráfico. Ele funciona com uma lista de redes, na qual cada rede contém uma estrutura que exibe, respectivamente, uma expressão lógica ou aritmética, uma chamada de bloco funcional, uma função, um programa, um salto ou uma instrução de retorno. Os comandos mais importantes são encontrados no menu de contexto (botão direito do mouse ou <Ctrl> + <F10>).

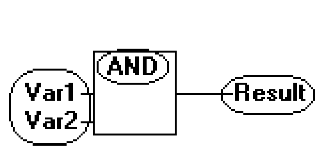
Considere a possibilidade de alternar a exibição de uma POU entre FBD e LD, no modo offline assim como no modo online. Também considere as possibilidades fornecidas pela caixa de diálogo de opções.Consultar, neste mesmo capítulo: Comentários, Linhas de Quebra,'Extras''Opções'.

Posições do Cursor em FBD

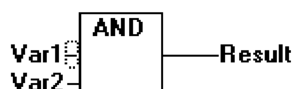
Todo o texto é uma posição possível do cursor. O texto selecionado está em um plano de fundo azul e agora pode ser alterado.

Também pode-se reconhecer a posição presente do cursor por um retângulo pontilhado. A seguir está uma lista de todas as posições possíveis de cursor com um exemplo.

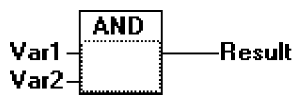
1)Campo de texto (posições possíveis de cursor emoldurados em preto):



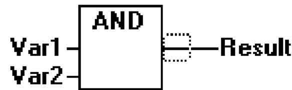
2)Entrada



3)Operador, função ou blocos funcionais:



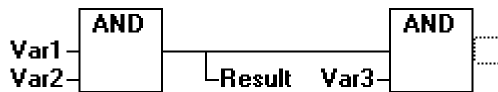
4)Saídas, se uma atribuição ou um salto vierem posteriormente:



5)Linha cruzada acima de uma atribuição, um salto ou uma instrução de retorno:



6) Atrás do objeto mais externo à direita de cada rede ("última posição do cursor", a mesma posição do cursor que foi usada para selecionar uma rede):



7)A linha cruzada diretamente na frente de uma atribuição:



Como ajustar o cursor no FBD

O cursor pode ser definido em uma determinada posição clicando com o mouse, ou com a ajuda do teclado.

Usando as setas de direção, pode-se ir para a posição do cursor mais próxima na direção selecionada a qualquer momento. Todas as posições do cursor, inclusive os campos de texto, podem ser acessadas desta forma. Se a última posição do cursor estiver selecionada, então, as setas <para cima> ou <para baixo> podem ser usadas para selecionar a última posição do cursor da rede anterior ou posterior.

Uma rede vazia contém apenas três pontos de interrogação "??? ". Clicando atrás deles, a última posição do cursor é selecionada.

'Inserir' 'Atribuição' em FBD

Símbolo:

Atalho: <Ctrl>+<A>

Este comando insere uma atribuição.

Dependendo da posição selecionada (consultar 'Posições do Cursor no FBD'), a inserção ocorre diretamente na frente da entrada selecionada (posição do cursor 2), diretamente após a saída selecionada (posição do cursor 4) ou no final da rede (posição do cursor 6).

Para uma atribuição inserida, uma seleção pode ser feita no texto "??? " e a atribuição pode ser substituída pela variável a ser atribuída. Para isso também pode ser usado o Assistente de Entrada. Para a possibilidade de inserir um endereço em vez do nome da variável considere a descrição da caixa de diálogo de opções.

Para inserir uma atribuição adicional a uma atribuição existente, use o comando 'Inserir' 'Saída'.

'Inserir' 'Salto' em FBD

Símbolo:

Atalho: <Ctrl>+<L>

Este comando insere um salto.

Dependendo da posição selecionada (consultar 'Posições do Cursor no FBD'), a inserção ocorre diretamente na frente da entrada selecionada (posição do cursor 2), diretamente após a saída selecionada (posição do cursor 4) ou no final da rede (posição do cursor 6).

Para um salto inserido, uma seleção pode ser feita no texto "???" e o salto pode ser substituído pelo rótulo a ser atribuído. Para isso também pode ser usado o Assistente de Entrada. Para a possibilidade de inserir um endereço em vez do nome da variável considere a descrição da caixa de diálogo de opções.

'Inserir' 'Retorno' em FBD

Símbolo:

Atalho: <Ctrl> + <R>

Este comando insere uma instrução de RETORNO.

Dependendo da posição selecionada (consultar 'Posições do Cursor no FBD'), a inserção ocorre diretamente na frente da entrada selecionada (posição do cursor 2), diretamente após a saída selecionada (posição do cursor 4), diretamente antes da linha cruzada selecionada (posição do cursor 5), ou no final da rede (posição do cursor 6).

'Inserir' 'Caixa (Box)' no FBD

Símbolo:

Atalho: <Ctrl>+<B>

Com este comando, operadores, funções, blocos funcionais e programas podem ser inseridos. Em primeiro lugar, é sempre inserido um operador "AND". Isso pode ser convertido por seleção ou sobrescrição do texto (“AND”) por outro operador, função, bloco funcional e programa. Pode-se selecionar a POU desejada usando o Assistente de Entrada (<F2>). Se o novo bloco selecionado tem outro número mínimo de entradas, essas serão anexadas. Se o novo bloco tem um número mais alto menor de entradas, as últimas entradas serão excluídas.

Em funções e blocos funcionais, os nomes formais das entradas e saídas são exibidas.

Em blocos funcionais existe um campo de instância editável acima da caixa. Se outro bloco funcional, que não é conhecido, é chamado, alterando o texto do tipo, uma caixa de operador com duas entradas e o tipo de dado é exibida. Se o campo de instância é selecionado, o assistente de entrada pode ser obtido por meio de <F2> com as categorias para seleção da variável.

A POU mais recente é inserida na posição selecionada:

- Se uma entrada é selecionada (posição do cursor 2), então, a POU é inserida na frente dessa entrada. A primeira entrada desta POU está vinculada a ramificação à esquerda da entrada selecionada. A saída da nova POU está vinculada a entrada selecionada.
- Se uma saída for selecionada (posição do cursor 4), então, a POU é inserida após essa saída. A primeira entrada da POU está conectada com a saída selecionada. A saída da nova POU está vinculada a ramificação com a qual a saída selecionada foi vinculada.
  Se uma POU, uma função ou um bloco funcional for selecionado (posição do cursor 3), o antigo elemento será substituído pela nova POU.
- Assim que possível, as ramificações serão conectadas da mesma maneira como estavam antes da substituição. Se o elemento antigo tinha mais entradas que o novo, então, as ramificações não anexadas serão excluídas. O mesmo se aplica para as saídas.
- Se um salto ou um retorno é selecionado, então, a POU será inserida antes desse salto ou retorno. A primeira entrada da POU está conectada com a ramificação à esquerda do elemento selecionado. A saída da POU está vinculada à ramificação à direita do elemento selecionado.
- Se a última posição do cursor de uma rede estiver selecionada (posição do cursor 6), então a POU será inserida após o último elemento. A primeira entrada da POU está vinculada à ramificação à esquerda da posição selecionada.

Todas as entradas de POU que não podem ser vinculadas receberão o texto "???". Este texto deve ser clicado e transformado na constante ou variável desejada.

Se houver uma ramificação à direita de uma POU inserida, a ramificação será atribuída à primeira saída da POU. Caso contrário, as saídas permanecem não atribuídas.

'Inserir' 'Entrada'

Símbolo:

Atalho: <Ctrl>+<U>

Este comando insere um operador de entrada. Com vários operadores, o número de entradas pode variar (por exemplo, ADD pode ter 2 ou mais entradas.)

Para estender um operador em uma entrada, é necessário selecionar a entrada na frente do qual se deseja inserir uma entrada adicional (posição do cursor 1); ou deve-se selecionar o operador propriamente dito (posição do cursor 3), se uma entrada mais baixo estiver para ser inserida (consultar 'Posições do cursor em FBD').

A entrada inserida é alocada com o texto "???". Este texto deve ser clicado e transformado na constante ou variável desejada.

Para isso também pode-se usar o Assistente de Entrada. Para inserir um endereço em vez do nome da variável considere a descrição da caixa de diálogo de opções.

Para a possibilidade de inserir um endereço em vez do nome da variável, considerar a descrição das opções de diálogo neste mesmo capítulo: Comentários, Linhas de quebra e 'Extras' 'Opções'.

'Inserir' 'Saída'

Símbolo:

Este comando insere uma atribuição adicional em uma atribuição existente. Esse recurso permite o posicionamento das assim chamadas atribuições “combs”; isto é, a atribuição de valor atualmente localizado na linha de diversas variáveis.

Ao selecionar a linha cruzada acima de uma atribuição (posição do cursor 5) (consultar 'Posições do cursor em FBD') ou a saída diretamente na frente dele (posição do cursor 4), então ali haverá outra atribuição inserida após aquelas que já estão lá.

Se a linha cruzada diretamente na frente de uma atribuição é selecionada (posição do cursor 4), outra atribuição será inserida na frente desta.

A saída inserida é alocada com o texto "???". Este texto deve ser clicado e alterado para a variável desejada. Para isso também pode-se usar o Assistente de Entrada. Para inserir um endereço em vez do nome da variável considere a descrição da caixa de diálogo Opções.

Para a possibilidade de inserir um endereço em vez do nome da variável, considerar a descrição das opções de diálogo neste mesmo capítulo: Comentários, Linhas de quebra e 'Extras' 'Opções'.

'Extras' 'Negar'

Símbolo:

Atalho: <Ctrl> + <N>

Com esse comando pode-se negar as entradas, saídas, saltos ou instruções de RETORNO. O símbolo para a negação é um pequeno círculo em uma conexão.

Se uma entrada estiver marcada (posição do cursor 2) (consultar 'Posições do cursor no FBD'), então essa entrada será negada.

Se uma saída for selecionada (posição do cursor 4), essa saída será negado.

Se um salto ou um retorno estiver marcado, a entrada desse salto ou retorno irá ser negada.

Uma negação pode ser cancelada por meio de outra negação no item.

'Extras' 'Set/Reset'

Símbolo:

Com esse comando pode-se definir saídas como Saídas Set ou Reset. Uma grade com Saída Set é exibida com [S], e uma grade com Saída Reset será exibida com [R].

Saída Set/Reset no FBD:

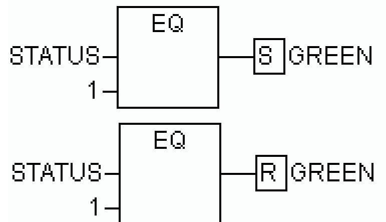


Figura 5-21. Saída Set/Reset no FBD

'Extras' 'Exibir' em FBD

Usando este comando para uma POU criada no Editor FBD pode-se escolher, se ela deve ser exibida em LD (lógica ladder) ou em Editor FBD (diagrama de blocos funcionais). Isso é possível tanto no modo offline quanto no online.

Abrir instância

Este comando corresponde ao comando de 'Projeto' 'Abrir instância'.

Ele está disponível no menu de contexto (<F2>) ou no menu 'Extras', se o cursor estiver posicionado sobre o nome de um bloco funcional em um editor de texto ou se a caixa de bloco de funções é selecionada em um editor gráfico.

Recortando, Copiando, Colando e Excluindo em FBD

Os comandos usados para **'Recortar'**, **'Copiar'**, **'Colar'** e **'Excluir'** são encontrados no item de menu **'Editar'**.

Se uma linha cruzada estiver marcada (posição do cursor 5) (consultar 'Posições do Cursor em FBD'), as atribuições, saltos ou RETORNOS localizados abaixo da linha cruzada serão recortados, excluídos ou copiados.

Se uma POU for selecionada (posição do cursor 3), então, o próprio objeto selecionado será recortado, excluído ou copiado, juntamente com todas as ramificações dependentes das entradas, com exceção da primeira ramificação (posição mais alta).

Caso contrário, a ramificação inteira localizada na frente da posição do cursor será recortada, excluída ou copiada.

Após copiar ou recortar, a parte excluída ou copiada fica localizada na área de transferência e agora pode ser colada, como desejado.

Para fazer isso, primeiro selecione o ponto de colagem. Pontos de colagem válidos incluem entradas e saídas.

Se uma POU foi carregada até a área de transferência (neste caso todas as ramificações conectadas, exceto a primeira, estão localizadas juntas na área de transferência), a primeira entrada é conectada com a ramificação antes do ponto de colagem.

Caso contrário, a ramificação inteira localizada na frente do ponto de colagem será substituída pelo conteúdo da área de transferência.

Em cada caso, o último elemento colado está conectado à ramificação localizada na frente do ponto de colagem.

Nota: o seguinte problema é resolvido recortando e colando... Um novo operador é inserido no meio de uma rede. A ramificação localizada à direita do operador agora está conectada com a primeira entrada, mas deve estar conectada com a segunda entrada. Então pode-se selecionar a primeira entrada e executar o comando **'Editar' 'Recortar'**. Após isso, pode-se selecionar a segunda entrada e executar o comando **'Editar' 'Colar'**. Dessa forma, a ramificação é dependente da segunda entrada.

O Diagrama de Blocos Funcionais no Modo Online

No diagrama de blocos funcionais, breakpoints somente podem ser definidos para as redes. Se um breakpoint foi definido para uma rede, o campo de número de rede será exibido em azul. O processamento então pára na frente da rede onde o breakpoint está localizado. Nesse caso, o campo de número de rede se tornará vermelho. Usando passo a passo, pode-se ir de rede a rede.

O valor atual é exibido para cada variável. Exceção: se a entrada para um bloco de funções é uma expressão, somente a primeira variável na expressão é monitorada.

O clique duplo em uma variável abre a caixa de diálogo para escrever uma variável. Aqui é possível alterar o valor presente da variável. No caso de variáveis booleanas, nenhuma caixa de diálogo aparece; essas variáveis são alternadas.

O novo valor ficará vermelho e permanecerá inalterado. Se o comando ‘Comunicação’ 'Escrever valores' for especificado, todas as variáveis são colocadas na lista selecionada e serão exibidas novamente em preto.

O controle de fluxo é iniciado com o comando ‘Comunicação’ 'Exibir controle de fluxo'. Usando o controle de fluxo, pode-se exibir os valores presentes que estão sendo executados nas redes através de linhas de conexão. Se as linhas de conexão não transmitem valores booleanos, o valor será exibido em um campo especialmente inserido. Se as linhas executam valores booleanos, então, eles serão sombreados em azul nos eventos que eles carregam TRUE. Através disto, pode-se acompanhar o fluxo de informações enquanto o CP está em execução.

Ao colocar o ponteiro do mouse brevemente acima de uma variável, então, o tipo, o endereço e o comentário referentes à variável serão exibidos em uma Tooltip.

O editor LD

Na figura é mostrado como uma POU escrita em LD aparece no editor MasterTool IEC:

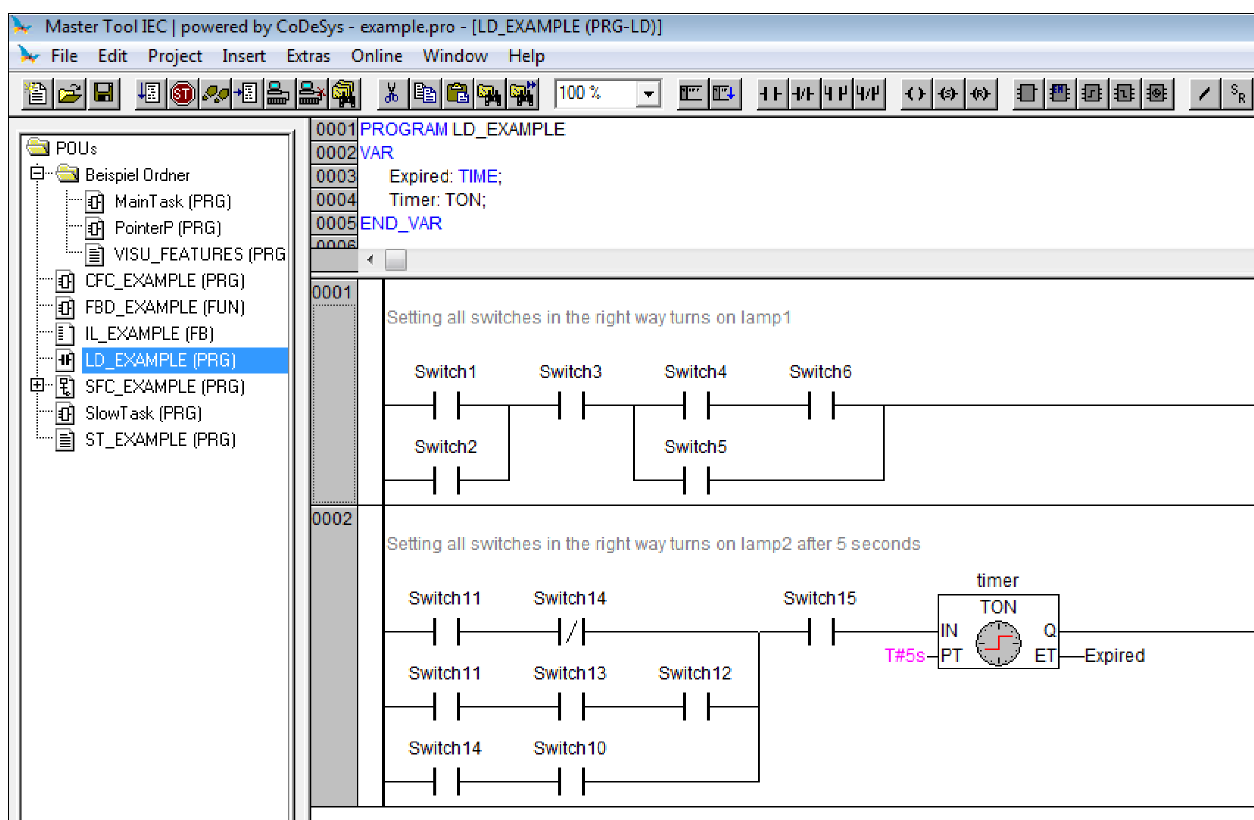


Figura 5-22. POU escrita em LD

Todos os editores para POUs consistem em uma parte de declaração e um corpo. Eles são separados por um divisor de tela.

O editor LD é um editor gráfico. Os comandos mais importantes são encontrados no menu de contexto (botão direito do mouse ou <Ctrl> + <F10>).

Para obter informações sobre os elementos, consultar o capítulo 2 Diagrama Ladder (LD). Considere as possibilidades referentes aos comentários, endereço de entrada e alternativa “look&feel”, fornecidos pela caixa de diálogo Opções (consultar capítulo 5).

Posições do cursor nos editores LD

Os seguintes locais podem ser posições do cursor, nas quais os blocos funcionais e acesso de programa podem ser tratados como contatos. POUs com entradas EN e outras POUs conectadas a eles são tratadas da mesma forma como no Diagrama de blocos funcionais.

Informações sobre como editar esta parte da rede podem ser encontradas neste mesmo capítulo: Editor FBD.

1. Campo de texto (posições de cursor possíveis com moldura em preto)



Figura 5-23. Posição do cursor - Campo de texto

2. Contato ou bloco funcional



Figura 5-24. Posição do cursor - Contato ou bloco funcional

3. Bobina



Figura 5-25. Posição do cursor - Bobina

4. A linha de conexão entre os contatos e as bobinas.

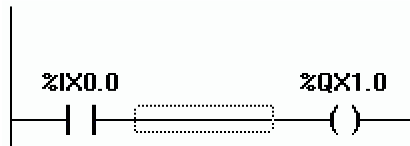


Figura 5-26. Posição do cursor - linha de conexão

O Diagrama LD usa os seguintes comandos de menu em uma forma especial...

Mover elementos ou nomes no Editor LD

Um elemento ou apenas o nome (nome de variável, endereço, comentário) de um elemento pode ser movido para uma posição diferente dentro de uma POU LD via “drag&drop”.

Para fazer isto selecione o elemento desejado (contato, bobina, blocos funcionais) e arraste-o mantendo a tecla do mouse pressionada - fora da posição atual. Desta forma, todas as possíveis posições em todas as redes da POU, para as quais o elemento pode ser movido, serão indicadas por retângulos preenchidos em cinza.

Mova o elemento a uma dessas posições e solte o mouse: o elemento será inserido na nova posição.

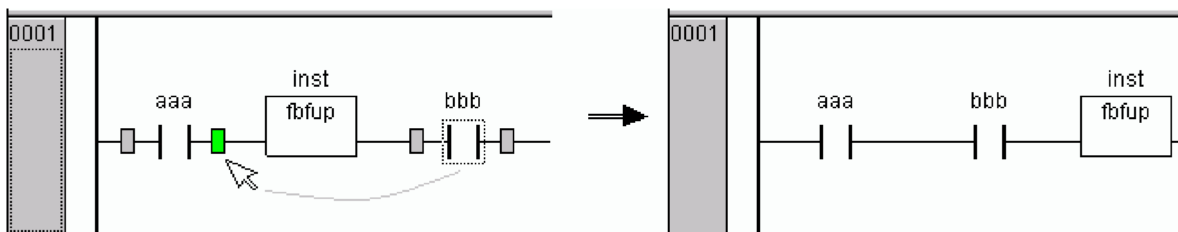


Figura 5-27. Inserção de elemento na nova posição

Se, entretanto, o elemento for movimentado para o nome (nome de variável) de outro elemento, o campo do nome será sombreado em verde. Então, ao soltar o mouse, o nome anterior será substituído pelo "arrastado". Se, além disso, endereço e comentário forem exibidos (Opções), a cópia também se aplicará a eles.

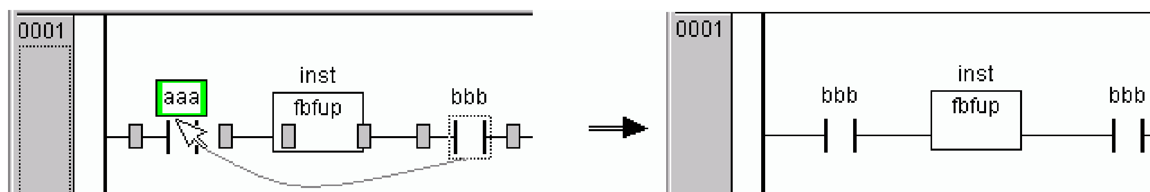


Figura 5-28. Movendo o elemento para o nome

'Inserir' 'Rede (antes)' no LD

Símbolo:

Este comando insere uma rede no editor LD. Se já houver redes, a nova será inserida antes da atual.

'Inserir' 'Rede (depois)' em LD

Símbolo:

Este comando insere uma rede no editor LD. Se já houver redes, a nova será inserida após a atual.

'Inserir' 'Contato' em LD

Símbolo:

Atalho: <Ctrl>+<K>

Use este comando no editor LD para inserir um contato na frente do local marcado na rede.

Se a posição marcada for uma bobina ou uma linha de conexão entre os contatos e as bobinas, então o novo contato será conectado em série para a conexão do contato anterior.

O contato é pré-definido com o texto "???". Pode-se clicar nesse texto e alterá-lo para a variável desejada ou para a constante desejada. Para isso também pode-se usar o Assistente de Entrada. Considerar a possibilidade de inserir um endereço em vez do nome da variável, se isso está configurado adequadamente em 'Opções de Blocos funcionais e Diagrama LD ('Extras' 'Opções').

Também na caixa de diálogo Opções, pode-se ativar as opções Comentários por Contato e Linhas para comentário de variável e reservar um determinado número de linhas para o nome da variável. Isso pode ser útil, se forem usados nomes de variáveis longos, para manter a rede curta.

Também considere a opção Rede com quebras de linha, a qual também pode ser ativada nas Opções de Diagrama LD.

'Inserir' 'Contato (negado)' em LD

Símbolo:

Atalho: <Ctrl>+<G>

Este comando insere um contato negado. O mesmo ocorre com os comandos 'Inserir' 'Contato' e 'Extras' 'Negar',que, em combinação, também podem ser usados para inserir um contato negado.

'Inserir' 'Contato paralelo' em LD

Símbolo:

Atalho: <Ctrl>+<R>

Use este comando no Editor LD para inserir um contato paralelo para a posição marcada na rede.

Se a posição marcada for uma bobina ou a conexão entre os contatos e as bobinas, então o novo contato será conectado em paralelo à conexão de contato anterior inteiro.

O contato é pré-definido com o texto "???". Pode-se clicar nesse texto e alterá-lo para a variável desejada ou para a constante desejada. Para isso também pode-se usar o Assistente de Entrada.

Para inserir endereços, linhas de quebra para nomes de variáveis e comentários por contato ou bobina, por favor, consultar a descrição da caixa de diálogo Opções (LD).

'Inserir' 'Contato paralelo (negado)' em LD

Símbolo:

Atalho: <Ctrl>+<D>

Este comando introduz um contato paralelo negado. O mesmo é verdadeiro para os comandos 'Inserir' 'Contato paralelo' e 'Extras' Negar', que, em combinação, igualmente podem ser usados para introduzir um contato paralelo negado.

'Inserir' 'Bobina' em LD

Símbolo:

Atalho: <Ctrl>+<L>

Pode-se usar este comando no editor LD para introduzir uma bobina paralelamente às bobinas precedentes.

Se a posição marcada é uma conexão entre os contatos e as bobinas, então a nova bobina estará introduzida como a última. Se a posição marcada é uma bobina, a seguir a nova bobina será introduzida diretamente acima dela.

A bobina é acompanhada pelo texto “???” como uma configuração padrão. Pode-se clicar sobre este texto e mudá-lo para variável desejada. Para isto pode-se igualmente usar o Assistente da Entrada.

Para inserir endereços, quebra de linha para nomes de variáveis e comentários por bobina veja, por favor, a descrição das Opções de diálogo do LD.

'Inserir' 'Bobina Set' em LD

Símbolo:

Atalho: <Ctrl>+<I>

Este comando insere uma bobina Set. O mesmo é verdadeiro para os comandos 'Inserir' 'Bobina' e 'Extras' 'Set/Reset' que, em combinação, também podem ser usados para obter uma bobina Set.

'Inserir' 'Bobina Reset' no LD

Símbolo:

Este comando insere uma Bobina Reset. O mesmo é verdadeiro para os comandos 'Inserir' 'Bobina' e 'Extras' 'Set/Reset' que, em combinação, também podem ser usados para obter uma Bobina Reset.

'Inserir' 'Blocos funcionais' em LD

Símbolo:

Atalho: <Ctrl>+<B>

Use esse comando para inserir um operador, um bloco funcional, uma função ou um programa como uma POU. Para isso, a conexão entre os contatos e bobinas ou bobina, deve ser marcada. A nova POU inicialmente tem a designação AND. Se desejar, pode-se alterar essa designação para outra. Para isso também pode-se usar o Assistente de entrada. Ambos, a POU padrão e a auto-definida estão disponíveis.

A primeira entrada para a POU é colocada na conexão de entrada, a primeira saída na conexão de saída; assim, essas variáveis, definitivamente, devem ser do tipo BOOL. Todas as outras entradas e saídas da POU são preenchidas com o texto “???”. Essas entradas podem ser transformadas em outras constantes, variáveis ou endereços. Para isso também pode-se usar o Assistente de entrada.

Para a possibilidade de inserir endereços, usar “linebreaks” em nomes de variáveis e comentários por contato (bobina, blocos funcionais), por favor, veja a descrição dos 'Diagramas de Blocos Funcionais e LD' (consultar neste capítulo: Comentários, Quebras de Linha, 'Extras' 'Opções').

'Inserir' POUs com Entrada EN

Para usar a rede LD como um CP para chamar outras POUs, então, deve-se mesclar uma POU com uma entrada EN. Tal POU está ligada em paralelo com as bobinas. Além desta POU, pode-se desenvolver outra rede, como no Diagrama de Blocos Funcionais. Pode-se encontrar os comandos para inserção em uma POU EN no item de menu **'Inserir' 'Inserir nos blocos'**.

Um operador, um bloco funcional, um programa ou uma função com entrada EN é executada da mesma forma que a POU correspondente no Diagrama de Blocos Funcionais, exceto que sua execução é controlada pela entrada EN. Esta entrada é anexada à linha de conexão entre bobinas e contatos. Se esta conexão carrega a informação "On", a POU será avaliada.

Se uma POU tiver sido criada uma vez com entrada EN, então, esta POU pode ser usada para criar uma rede. Isso significa que dados provenientes de operadores habituais, funções e blocos funcionais podem fluir em uma POU EN e uma POU EN pode carregar dados para essas POUs habituais.

Para programar uma rede no editor LD, tal como no FBD, é necessário primeiro inserir um operador EN em uma nova rede. Posteriormente, a partir da presente POU, pode-se continuar a editar, a partir da rede, como no editor FBD. Uma rede assim formada será executada como uma rede correspondente em FBD.

'Inserir' 'Caixa com EN' em LD

Símbolo:

Use este comando para inserir um bloco funcional, um operador, uma função ou um programa com entrada EN em uma rede LD.

A posição marcada deve ser a conexão entre os contatos e bobinas (posição do cursor 4) ou uma bobina (posição do cursor 3). A nova POU é inserida em paralelo às bobinas e embaixo delas; ela contém inicialmente a designação "AND". Se desejar, pode-se alterar essa designação para outra. Para isso também pode ser usado o Assistente de Entrada.

'Inserir' 'Inserir em blocos' no LD

Com esse comando pode-se inserir elementos adicionais em uma POU que já tenha sido inserida (também uma POU com entrada EN). Os comandos abaixo deste item de menu podem ser executados nas mesmas posições de cursor que os comandos correspondentes no Diagrama de Blocos Funcionais.

Com **Entrada**, pode-se adicionar uma nova entrada para a POU.

Com **Saída** pode-se adicionar uma nova saída para a POU.

Com **POU**, se insere uma nova POU. O procedimento é semelhante ao descrito em 'Inserir' 'POU'.

Com **Atribuição**, pode-se inserir uma atribuição para uma variável. Primeiramente, isso é mostrado por três pontos de interrogação “???”, que devem ser substituídos pela variável desejada. O Assistente de Entrada está disponível para essa finalidade.

'Inserir' 'Detecção de borda de subida' no LD

Símbolo:

Este comando insere um bloco funcional R_TRIG, que serve para detectar uma borda de subida (FALSE para TRUE) no sinal de entrada. O mesmo é verdadeiro para o comando 'Inserir' 'Bloco funcional' que pode ser usado para inserir qualquer bloco funcional disponível.

'Inserir' 'Detecção de borda de descida' no LD

Símbolo:

Este comando insere um bloco funcional F_TRIG, que serve para detectar uma borda de descida (TRUE -> FALSE) no sinal de entrada. O mesmo é verdadeiro para o comando 'Inserir' 'Bloco funcional' que pode ser usado para inserir qualquer bloco funcional disponível.

'Inserir' 'Temporizador (TON)' no LD

Símbolo:

Este comando insere um bloco funcional de temporização do tipo TON. Isso permite obter um retardo na energização. Para a inserção, o mesmo é verdadeiro para o comando 'Inserir' 'Bloco funcional', que também pode ser usado para inserir um módulo TON.

'Inserir' 'Salto' no LD

Com esse comando que pode-se inserir um salto paralelo no editor LD selecionado, em paralelo, no final das bobinas anteriores. Se a linha de entrada fornece o valor "On", então, o salto será executado para o rótulo indicado.

A posição marcada deve ser a conexão entre os contatos e as bobinas ou uma bobina.

O salto é apresentado com o texto “???”. Pode-se clicar nesse texto e fazer uma alteração desejada no rótulo.

'Inserir' 'Retorno' no LD

No editor LD, pode-se usar esse comando para inserir uma instrução retorno em paralelo ao final das bobinas anteriores. Se a linha de entrada fornece o valor "On", então, o processamento da POU desta rede é interrompido.

A posição marcada deve ser a conexão entre os contatos e as bobinas ou uma bobina.

'Extras' 'Colar depois' no LD

Use este comando no editor LD para colar o conteúdo da área de transferência como contato “em série” após da posição marcada. Este comando só é possível se o conteúdo da área de transferência e a posição marcada são redes compostas de contatos.

'Extras' Colar abaixo' no LD

Use este comando no editor LD para inserir o conteúdo da área de transferência como contato abaixo da posição marcada. Este comando só é possível se o conteúdo da área de transferência e a posição marcada são redes compostas de contatos.

'Extras' 'Colar acima' no LD

Use este comando no editor LD para inserir o conteúdo da área de transferência como contato acima da posição marcada. Este comando só é possível se o conteúdo da área de transferência e a posição marcada são redes compostas de contatos.

'Extras' 'Negar' no LD

Símbolo:

Atalho: <Ctrl>+<N>

Use esse comando para negar um contato, uma bobina, uma instrução de salto ou retorno, ou uma entrada ou saída de POUs EN na posição do cursor atual.

Entre os parênteses da bobina ou entre as linhas retas do contato, uma barra aparecerá ((/) ou |/|). Se houver saltos, retornos, ou as entradas ou saídas de POUs EN, um círculo pequeno aparecerá em uma conexão, assim como no Editor FBD.

A bobina agora escreve o valor negado da conexão de entrada na respectiva variável Booleana e um contato negado alterna o status da entrada para a saída, se a respectiva variável booleana carrega o valor FALSE.

Se um salto ou um retorno estiver marcado, a entrada desse salto ou retorno será negada.

Uma negação pode ser cancelada por meio de uma nova negação.

'Extras' 'Set/Reset' no LD

Símbolo:

Esse comando, em uma bobina, configura-a como uma Bobina Set. Esta bobina nunca sobrescreve o valor TRUE na respectiva variável Booleana. Isso significa que depois de ter definido o valor dessa variável para TRUE, ela sempre permanecerá TRUE. Uma Bobina Set é designada com um "S" no símbolo de bobina.

Ao executar esse comando novamente, a bobina é configurada como Bobina Reset. Uma bobina deste tipo nunca sobrescreve o valor FALSE na respectiva variável Booleana. Isso significa que depois de ter definido o valor dessa variável para FALSE, ela sempre permanecerá FALSE. Uma Bobina Reset é designada com um "R" no símbolo de bobina.

Executando esse comando repetidamente, a bobina irá alternar entre bobina Set, Reset e normal.

O Diagrama LD no modo online

No modo online, os contatos e bobinas no Diagrama LD que estão no estado "On" ficam em azul. Da mesma forma, todas as linhas através do qual o "On" é transferido também ficam em azul. Nas entradas e saídas de blocos funcionais, os valores das variáveis correspondentes são indicados.

Breakpoints somente podem ser definidos em redes; usando as etapas, pode-se ir de rede a rede.

Colocando o ponteiro do mouse sobre uma variável, então, o tipo, o endereço e o comentário referentes à variável serão exibidos em uma tooltip.

O editor SFC

A figura a seguir mostra como uma POU escrita em SFC aparece no Editor MasterTool IEC:

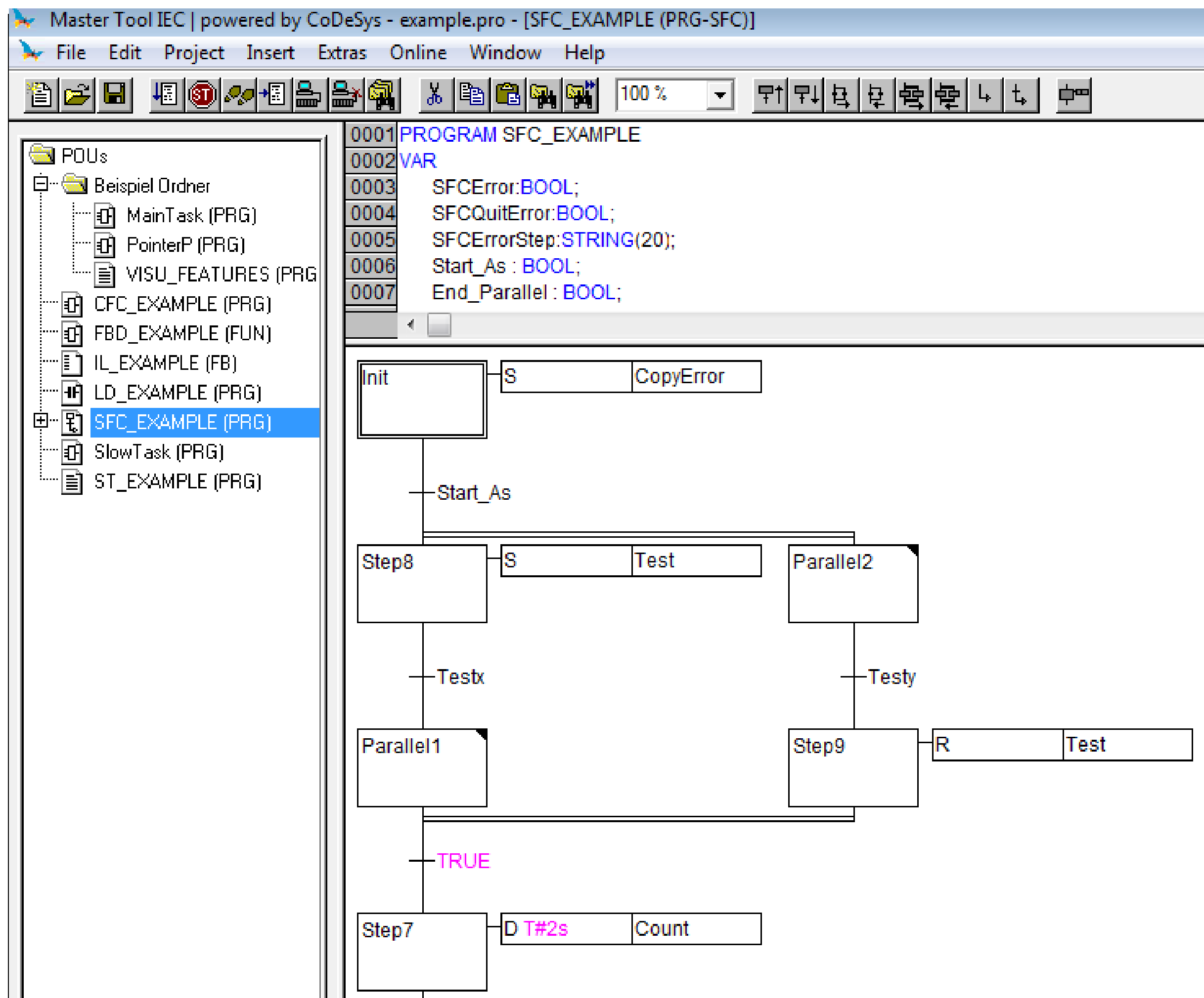


Figura 5-29. POU escrita em SFC

Todos os editores para POUs consistem em uma parte de declaração e um corpo. Eles são separados por um divisor de tela.

O Editor SFC é um editor gráfico. Os comandos mais importantes são encontrados no menu de contexto (botão direito do mouse ou <Ctrl>+<F10>). Tooltips mostram tanto no modo offline quanto no online e no estado ampliado, os nomes completos ou expressões de passos, transições, saltos, rótulos de saltos, qualificadores ou ações associadas.

Para obter informações sobre SFC, consultar o capítulo 2: Seqüenciamento Gráfico de Funções.

O editor para a referida linguagem deve concordar com as particularidades do SFC. Em referência a elas, os seguintes itens de menu serão úteis.

Marcando blocos no SFC

Um bloco marcado é um grupo de elementos do SFC colocados entre um retângulo pontilhado.

Pode-se selecionar um elemento (um passo, uma transição ou um salto), apontando o mouse neste elemento e pressionando o botão esquerdo do mouse, ou pode-se usar as teclas de seta. Para marcar um grupo de vários elementos, pressione <Shift> para um bloco já marcado e selecione o elemento no canto inferior esquerdo ou direito do grupo. A seleção resultante é o menor grupo coeso de elementos que inclua esses elementos.

Considerar que um passo só pode ser excluído junto com a transição anterior ou subseqüente!

'Inserir' 'Transição de passo (antes)'

Símbolo:

Atalho: <Ctrl>+<T>

Este comando insere um passo no editor SFC seguido por uma transição em frente ao bloco marcado.

Pode-se selecionar e substituir automaticamente o nome do passo especificado; „Step-<x>"por outra string, também nome de transição.

Nota: ao renomear um passo, considere que nenhum comentário pode ser adicionado. Exemplo: „Step_xy (* contador *)" não é permitido.

'Inserir''Transição de passo (depois)'

Símbolo:

Atalho: <Ctrl>+<E>

Este comando insere um passo no editor SFC seguido por uma transição após a primeira transição no bloco marcado.

Referente ao nome do passo e transição, por favor, veja acima 'Inserir 'Passo-Transição (antes)'.

Excluir Passo e Transição

Um passo só pode ser excluído junto com a transição anterior ou subseqüente. Para essa finalidade colocar um quadro de seleção ao redor do passo e transição e escolher o comando 'Editar' 'Excluir' ou pressione a tecla <Del>.

'Inserir' 'Ramificação alternativa (direita)'

Símbolo:

Atalho: <Ctrl>+<A>

Este comando insere uma ramificação alternativa no editor SFC como uma ramificação à direita do bloco marcado. Para isso o bloco marcado deve tanto começar quanto terminar com uma transição. A nova ramificação, então, é composta de uma transição.

'Inserir' 'Ramificação alternativa (esquerda)'

Símbolo:

Este comando insere uma ramificação alternativa no editor SFC como uma ramificação à esquerda do bloco marcado. Para isso o bloco marcado deve tanto começar quanto terminar com uma transição. A nova ramificação, então, é composta de uma transição.

'Inserir' 'Ramificação paralela (direita)'

Símbolo:

Atalho: <Ctrl>+<L>

Este comando insere uma ramificação paralela no editor SFC como uma ramificação à esquerda do bloco marcado. Para isso o bloco marcado deve tanto começar quanto terminar com uma transição. A nova ramificação, então, é composta de um passo. Para permitir saltos para as ramificações paralelas que foram criadas, elas devem ser fornecidas com um rótulo de salto.

'Inserir' 'Ramificação paralela (esquerda)'

Símbolo:

Este comando insere uma ramificação paralela no editor SFC como uma ramificação à esquerda do bloco marcado. Para isso o bloco marcado deve tanto começar quanto terminar com um passo. A nova ramificação, então, é composta de um passo. Para permitir saltos para as ramificações paralelas que foram criadas, elas devem ser fornecidas com um rótulo de salto (consultar 'Extras' 'Adicionar rótulo às ramificações paralelas').

'Inserir' 'Salto'

Este comando insere um salto no editor SFC no final da ramificação, ao qual pertence o bloco marcado. Para isso a ramificação deve ser uma ramificação alternativa.

A string de texto inserida 'Step', no salto inserido, pode ser selecionada e substituída pelo nome do passo ou pelo rótulo de salto de uma ramificação paralela (escopo do salto).

Referente ao nome do passo e transição, por favor, veja acima 'Inserir 'Passo-Transição (antes)'.

'Inserir''Salto de transição'

Símbolo:

Este comando insere uma transição no editor SFC, seguida por um salto ao final da ramificação selecionada. Para isso a ramificação deve ser uma ramificação paralela.

A string de texto 'Step' inserida, no salto inserido, pode ser selecionada e substituída pelo nome do passo ou pelo rótulo de salto de uma ramificação paralela (escopo do salto).

Referente ao nome do passo e transição, por favor, veja acima 'Inserir 'Passo-Transição (antes)'.

'Inserir''Adicionar ação de entrada'

Com esse comando pode-se adicionar uma ação de entrada a um passo. Uma ação de entrada só é executada uma vez, logo após o passo tornar-se ativo. A ação de entrada pode ser implementada em uma linguagem a sua escolha.

Um passo com uma ação de entrada é designado por um "E" no canto inferior esquerdo.

'Inserir''Adicionar ação de saída'

Com esse comando pode-se adicionar uma ação de saída a um passo. Uma ação de saída só é executada uma vez, antes que o passo seja desativado. A ação de saída pode ser implementada em uma linguagem a sua escolha.

Um passo com uma ação de saída é designado por um "X" no canto inferior direito.

'Extras' 'Colar Ramificação Paralela (direita)'

Este comando cola o conteúdo da área de transferência como uma ramificação paralela à direita do bloco marcado. Para isso, o bloco marcado deve tanto começar quanto terminar com um passo. O conteúdo da área de transferência, da mesma forma, deverá ser um bloco de SFC que comece e termine com um passo.

'Extras' 'Adicionar rótulo para ramificação paralela (direita)'

A fim de fornecer uma ramificação paralela recém-inserida com um rótulo de salto, a transição que ocorrer antes da ramificação paralela deve ser marcada e o comando 'Adicionar rótulo à ramificação paralela' deve ser executado. Nesse ponto, à ramificação paralela será dado um nome padrão, que consiste de "Parallel" e um número de série acrescentado, o qual pode ser editado conforme as regras de nomes de identificador. No exemplo a seguir, "Parallel" foi substituído por "Par_1_2" e o salto para a transição "Final" foi dirigido para esse rótulo de salto.

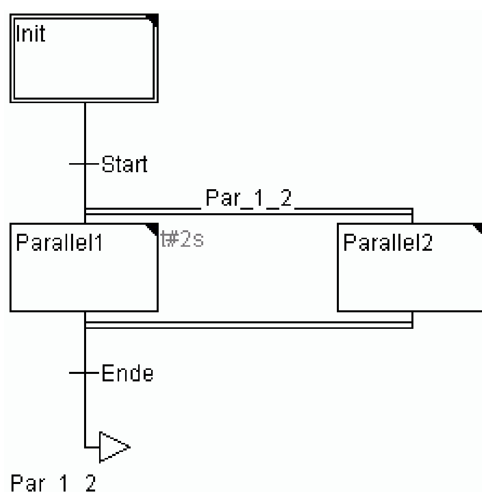


Figura 5-30. Rótulo para ramificação paralela

Excluir um rótulo

Um rótulo de salto pode ser excluído, excluindo o nome do rótulo.

'Extras' 'Colar após'

Este comando cola o bloco SFC na área de transferência após o primeiro passo ou a primeira transição do bloco marcado (copiando normal e colando-o em frente do bloco marcado). Isso será agora executado se a estrutura resultante do SFC estiver correta, conforme as normas de linguagem.

'Extras' 'Zoom ação/transição'

Atalho: <Alt>+<Enter>

A ação do primeiro passo do bloco marcado ou do corpo de transição da primeira transição do bloco marcado é carregada no editor, na respectiva linguagem na qual ela foi escrita. Se a ação ou o corpo de transição estiver vazio, a linguagem na qual ele foi escrito deve ser selecionada.

Considere que a condição de transição que está escrita dentro da janela do editor terá precedência sobre uma condição que pode ser gravada diretamente na marca de transição. Exemplo: Se aqui i>100, a condição de transição será FALSE, embora TRUE tenha sido inserido na marca!

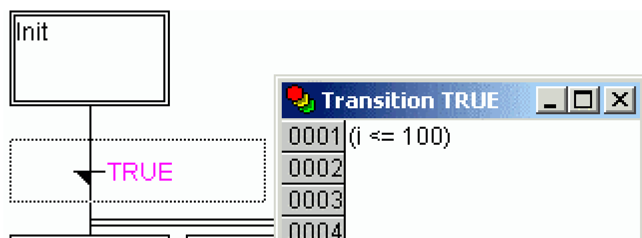


Figura 5-31. Zoom ação/transição

'Extras' 'Limpar ação/transição'

Com esse comando pode-se excluir as ações do primeiro passo do bloco marcado ou do corpo das transições da primeira transição.

Se, durante um passo, for implementada apenas a ação, a ação de entrada ou ação de saída, então a mesma será excluída pelo comando. Caso contrário, uma caixa de diálogo aparece e pode-se selecionar qual ação ou ações devem ser excluídas.

Se o cursor estiver localizado na ação de um passo IEC, então somente essa associação será excluída. Se um passo IEC com uma ação associada for selecionado, essa associação será excluída. Durante um passo IEC com várias ações, uma caixa de diálogo de seleção será exibida.

'Extras' 'Atributos de passo'

Com esse comando pode-se abrir uma caixa de diálogo na qual pode-se editar os atributos para o passo marcado.

Caixa de diálogo para Edição de atributos de passo:

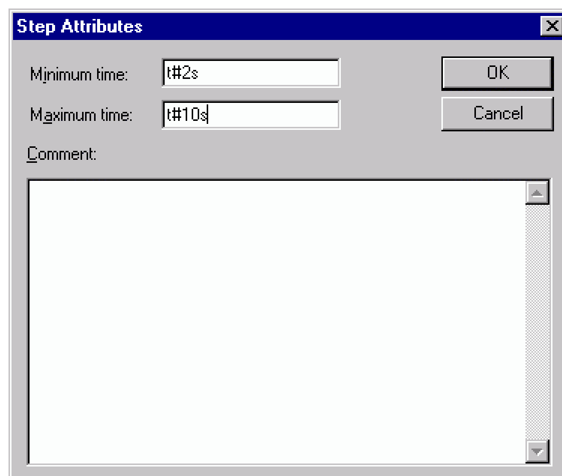


Figura 5-32. Caixa de diálogo para Edição de atributos de passo

Pode-se utilizar três entradas diferentes na caixa de diálogo de atribuição de passo. Em **Tempo Mínimo**, deve-se inserir o valor de tempo mínimo que o processamento deste passo deve tomar. Sob o **Tempo Máximo**, deve-se inserir o valor de tempo máximo que o processamento deste passo deve tomar. Observe que as entradas são do tipo **TIME**, portanto, se usa uma constante de TEMPO (isto é, T#3s) ou uma variável do tipo TIME.

As configurações de tempo também são acessíveis na caixa de diálogo 'Extras' 'Visão geral do tempo'.

Em **Comentários**, pode-se inserir um comentário para o passo. Na caixa de diálogo 'Opções do SFC', que pode ser aberta em 'Extras' 'Opções', pode-se, então, inserir comentários ou a configuração de tempo é exibida para as etapas no Editor SFC. À direita, ao lado do passo, o comentário ou a configuração de tempo será exibida.

Se o tempo máximo for excedido, os sinalizadores SFC são definidos e o usuário pode avaliá-los.

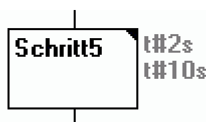


Figura 5-33. Exemplo de atributos de passo

O exemplo mostra um passo cuja execução deve durar pelo menos dois e no máximo, dez segundos. No modo online há, além dessas duas vezes, uma exibição de há quanto tempo o passo já está ativo.

'Extras' 'Visão geral do tempo'

Com esse comando pode-se abrir uma janela na qual pode-se editar as configurações de tempo para os seus passos SFC.

Na visão geral dos limites de tempo, todos os passos de sua POU SFC são exibidos. Se foi inserido um limite de tempo para um passo, o mesmo é exibido à direita do passo (primeiro o limite inferior, depois o limite superior). Também pode-se editar os limites de tempo. Para fazer isso, clique no passo desejado na visão geral. O **nome do passo** é então mostrado abaixo na janela. Vá para o campo **Tempo Mínimo** ou **Tempo Máximo** e insira ali o limite de tempo desejado. Ao fechar a janela com **OK**, todas as alterações serão armazenadas.

Visão Geral dos Limites de Tempo para uma POU SFC:

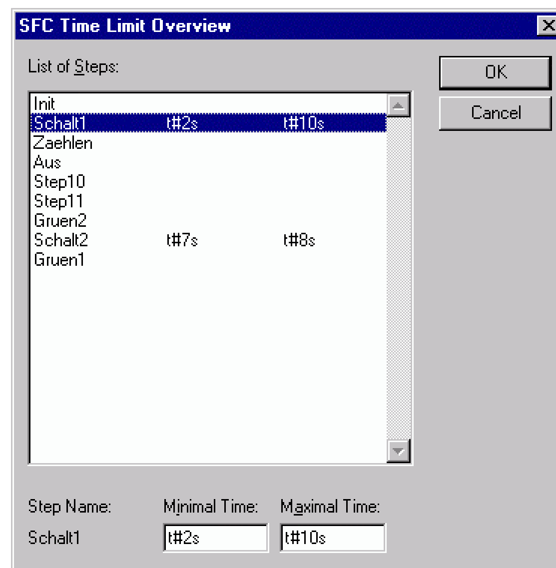


Figura 5-34. Visão Geral dos Limites de Tempo para uma POU SFC

No exemplo, os passos 2 e 6 possuem um limite de tempo. Shift1 dura pelo menos dois e, no máximo, dez segundos. Shift2 dura pelo menos sete e, no máximo, oito segundos.

'Extras' 'Opções'

Com esse comando obtém-se uma caixa de diálogo na qual pode-se definir diferentes opções para a sua POU SFC.

Em **Altura do Passo**, pode-se definir quantas linhas um passo do SFC pode ter em seu editor. 4 é o padrão definindo aqui. Em **Largura do Passo**, pode-se inserir quantas colunas deve ter um passo. 6 é o padrão definindo aqui.

Também pode ser pré-definida a exibição no passo, isto é, quais dos atributos definidos em 'Extras' 'Atributos de passo' devem ser exibidos ao lado do passo. Escolha Comentários, Limites de Tempo ou nenhum dos dois:

Se “Nenhum” está selecionado, o comentário definido e os limites de tempo podem ser mostrados em uma tooltip, a qual aparece quando o cursor está posicionado no canto inferior direito da caixa de passo.

Se “Comentário” está selecionado, o comentário definido e os limites de tempo podem ser mostrados em uma tooltip, a qual aparece quando o cursor está posicionado no canto superior direito da caixa de passo.

Se “Limites de Tempo” está selecionado, o comentário definido e os limites de tempo podem ser mostrados em uma tooltip, a qual aparece quando o cursor está posicionado no canto inferior direito da caixa de passo.

Caixa de diálogo para Opções SFC:

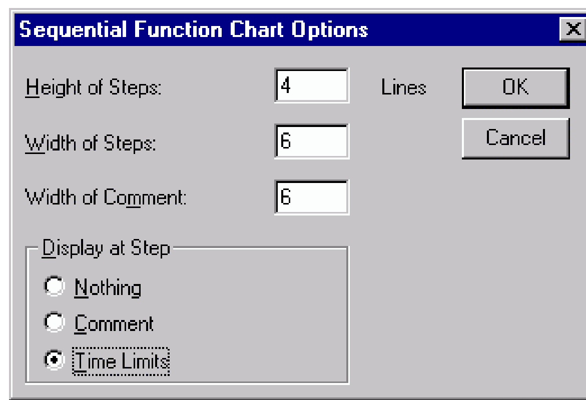


Figura 5-35. Caixa de diálogo para Opções SFC

'Extras' 'Ações associadas'

Com este comando ações e variáveis Booleanas podem ser associadas a passos IEC.

À direita de e próximo à etapa IEC, uma caixa adicional dividida está anexada, para a associação de uma ação. Ela é pré-definida no campo à esquerda com o qualificador "N" e o nome "Ação". Ambas as pré-definições podem ser alteradas. Para isso pode-se usar o Assistente de entrada.

No máximo, nove ações podem ser atribuídas a um passo IEC.

Novas ações para passos IEC são criadas no Organizador de objetos para uma POU SFC com o comando **'Projeto' 'Acrescentar ação'**.

'Extras' 'Usar passos IEC'

Símbolo:

Se esse comando está ativado (indicado por um sinal de verificação na frente do item de menu e um símbolo impresso na barra de ferramentas), então os passos IEC serão inseridos, em vez dos passos simplificados, após a inserção de transições de passo e ramificações paralelas.

Se essa opção está selecionada, o passo de inicialização é definido como um passo IEC ao criar uma nova POU SFC.

Essas configurações são salvas no arquivo-ini e são restauradas quando **MasterTool IEC** é iniciado novamente.

SFC no modo online

Com o Editor SFC no modo online, os passos ativos no momento serão exibidos em azul. Se definido em 'Extras' 'Opções', então, o gerenciamento de tempo é descrito junto com os passos. Sob os limites inferior e superior definidos, um terceiro indicador de tempo aparecerá, a partir do qual pode-se verificar a quanto tempo o passo está ativo.

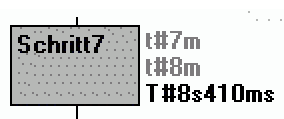


Figura 5-36. Detalhe do tempo que o passo está ativo

SFC com um passo Ativo (Shift1) e um Breakpoint (Step10):

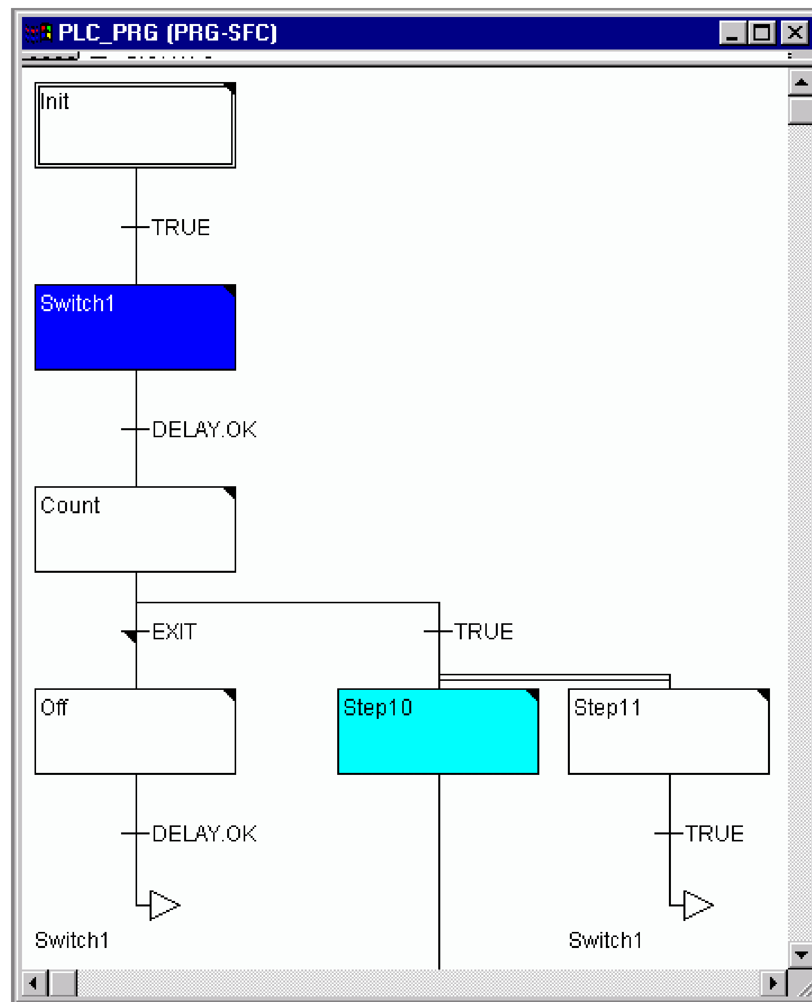


Figura 5-37. SFC com um passo Ativo (Shift1) e um Breakpoint (Step10)

Na imagem acima o passo representado já está ativa há 8 segundos e 410 milissegundos. No entanto, o passo deverá estar ativo por pelo menos 7 minutos antes que ele seja deixado.

Com **'Comunicação' 'Alternar breakpoint'** um breakpoint pode se definido em um passo ou em uma ação nos locais permitidos pela linguagem em uso. O processamento, então, interrompe antes da execução deste passo, ou antes, do local da ação no programa. Passos ou locais do programa onde um breakpoint é definido são marcados em azul claro.

Se vários passos estão ativos em uma ramificação paralela, o passo ativo cuja ação será processada a seguir será exibido em vermelho.

Se Passos IEC estiverem sendo usados, todas as ações ativas no modo online serão exibidos em azul.

Com o comando **'Comunicação' 'Sobrepasso'** ele é passado sempre para o próximo passo em que a ação é executada. Se o local atual for:

- Um passo do processamento linear de uma POU ou um passo na ramificação paralela extrema direita de uma POU, a execução retorna a partir da POU SFC para o chamador. Se a POU é o programa principal, o próximo ciclo será iniciado.
- Um passo em uma ramificação paralela mais à direita, a execução salta para o passo ativo na próxima ramificação paralela.
- O último breakpoint local em uma ação ALTUS, a execução salta para o chamador do SFC.
- O último breakpoint no local dentro de uma ação IEC, a execução salta para o chamador do SFC.

- a última posição do breakpoint dentro de uma ação de entrada ou ação de saída, a execução salta para o próximo passo ativo.

Com **‘Comunicação’ 'Passo a passo'** até ações podem ser “passadas”. Se uma entrada, saída ou ação IEC devem ser ignoradas, um breakpoint deve ser definido ali. Dentro das ações, toda a funcionalidade de depuração do editor correspondente está disponível para o usuário.

Ao posicionar o cursor do mouse por um curto período de tempo em uma variável no Editor de declaração, o tipo, o endereço e o comentário da variável serão exibidos em uma **tooltip**.

Considerar: ao renomear um passo e executar uma alteração online enquanto esse passo está ativo, o programa será interrompido em status indefinido!

Ordem de processamento de elementos em uma seqüência:

1. Primeiro todas as memórias de bloco de controle de ação nas ações IEC que são usadas nessa seqüência são zeradas (no entanto, não as memórias de ações IEC que são chamadas de dentro de ações).
2. Todos os passos são testados na ordem em que eles assumem na seqüência (de cima para baixo) e da esquerda para a direita para determinar se o requisito para execução da ação de saída é fornecido, e isso é executado se esse for o caso.
3. Todos os passos são testados na ordem em que eles assumem a seqüência para determinar se o requisito para a ação de entrada é fornecido e isso será executado se esse for o caso.
4. Para todos os passos, o seguinte é feito na ordem em que eles assumem a seqüência:

- - Se aplicável, o tempo decorrido é copiado para a correspondente variável de passo.
- - Se aplicável, qualquer tempo limite é testado e as memórias de erro SFC são atendidas como requerido.
- - Para passos não-IEC, a ação correspondente agora é executada.

5. Ações IEC usadas em seqüência são executadas em ordem alfabética. Isso é feito em duas passagens através da lista de ações. Na primeira passagem, todas as ações IEC que estão desativadas no ciclo atual são executadas. Na segunda passagem, todas as ações IEC que estão ativas no ciclo atual são executadas.
6. Transições são avaliadas: se o passo do ciclo atual estava ativo, a transição seguinte retorna TRUE (e, se aplicável, o mínimo tempo ativo já expirou), então o passo seguinte é ativado.

O que segue deve ser observado referente à implementação de ações...

Pode ocorrer que uma ação seja realizada várias vezes em um ciclo porque ela está associada a várias seqüências (por exemplo, um SFC poderia ter duas ações IEC, A e B, que são ambas implementadas no SFC e ambas chamadas ações IEC C; então, as ações IEC A e B podem ambas estar ativas no mesmo ciclo e, além disso, em ambas as ações IEC, a ação C pode estar ativa; então, C deve ser chamada duas vezes).

Se a mesma ação IEC for usada simultaneamente em diferentes níveis de um SFC, isso pode levar a efeitos indesejados devido à seqüência de processamento descrita acima. Por esse motivo, uma mensagem de erro é emitida nesse caso. Isso, possivelmente, pode ocorrer durante o processamento dos projetos criados em versões mais antigas do MasterTool IEC.

Nota: na monitoração de expressões em transições (por exemplo, A AND B), somente o “valor total” da transição é exibido.

O editor CFC

Esta é a aparência de um bloco que foi produzido usando o editor CFC:

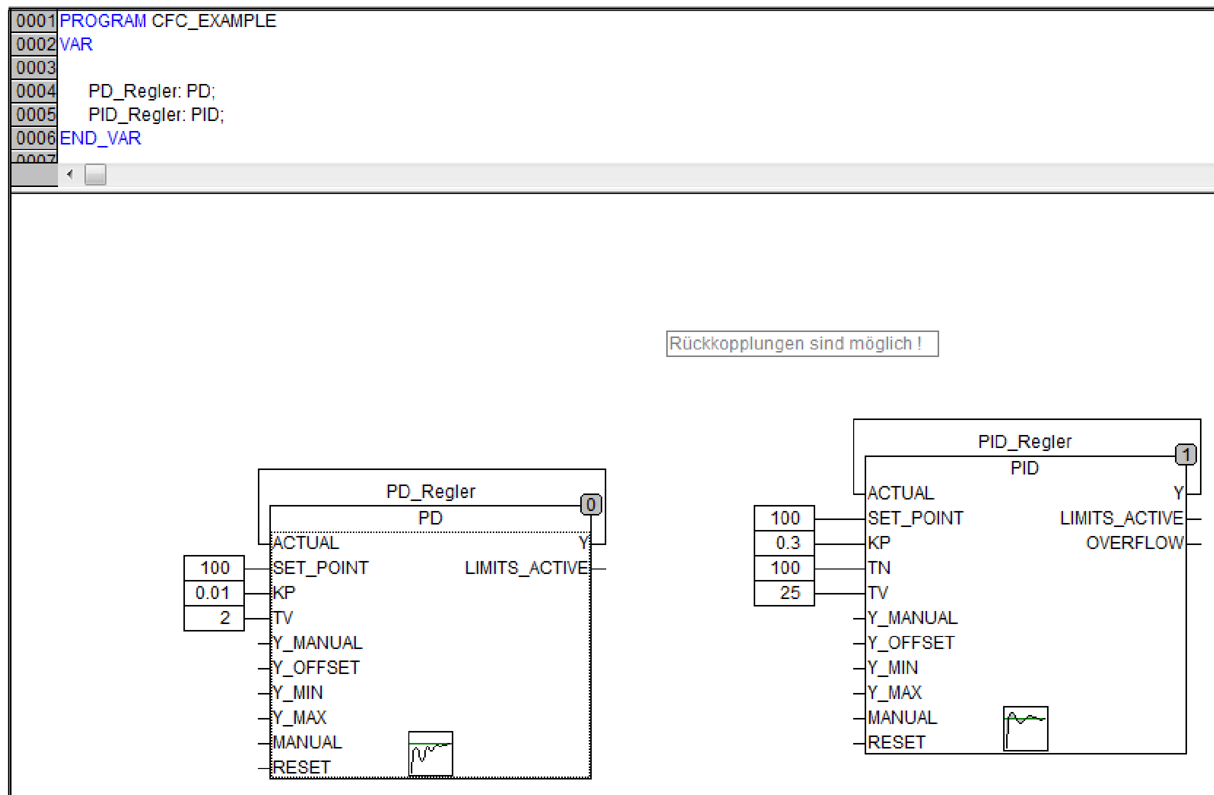


Figura 5-38. Editor CFC

Nenhuma grade “snap” é usada para o editor de gráfico contínuo de função, então os elementos podem ser colocados em qualquer lugar. Elementos de lista de processamento seqüencial incluem caixas de entrada, saída, salto, rótulo, retorno e comentários. As entradas e saídas desses elementos podem ser conectadas arrastando uma conexão com o mouse. A linha de conexão será chamada automaticamente. A linha de conexão possível mais curta é desenhada tendo em conta as conexões existentes. As linhas de conexão são automaticamente ajustadas quando os elementos são movidos. Se houver o caso onde uma linha de conexão não pode estabelecer-se simplesmente por causa da falta de espaço, uma linha vermelha será mostrada entre a entrada e saída associada. Esta linha será convertida em uma linha de conexão, assim que houver espaço disponível.

Uma vantagem do gráfico contínuo de função em relação ao editor de diagrama de blocos funcionais usual é o fato de que os caminhos de comentários podem ser inseridos diretamente.

Os comandos mais importantes podem ser encontrados no menu de contexto.

Posições do cursor no CFC

Cada texto é uma possível posição do cursor. O texto selecionado é sombreado em azul e pode ser modificado.

Em todos os outros casos a posição do cursor atual é mostrada por um retângulo formado por pontos. A seguir está uma lista de todas as posições possíveis do cursor com exemplos:

1. Troncos de caixa de elementos, entrada, saída, salto, rótulo, retorno e comentários.

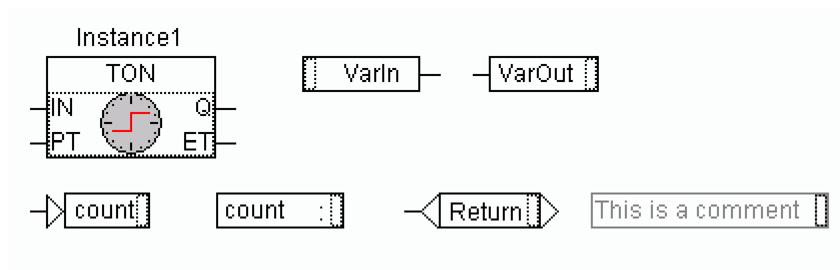


Figura 5-39. Posições do cursor em CFC - 1

2. Campos de texto para a caixa de elementos, entrada, saída, salto, rótulo, retornos e comentários, bem como campos de texto para o marcador de conexão.

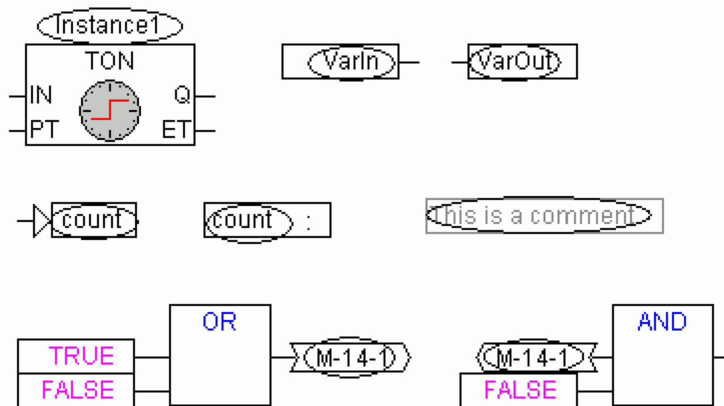


Figura 5-40. Posições do cursor em CFC - 2

3. Entradas para a caixa de elementos, entrada, saída, salto e retornos.

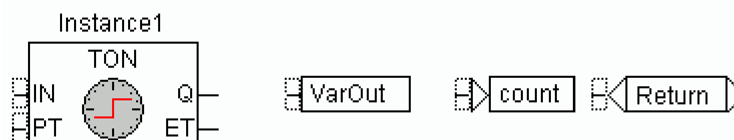


Figura 5-41. Posições do cursor em CFC - 3

4. Saídas para a caixa de elementos e entrada:

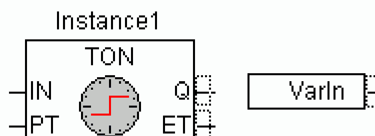


Figura 5-42. Posições do cursor em CFC - 4

'Inserir' 'Caixa' no CFC

Símbolo:

Atalho: <Ctrl>+<B>

Este comando pode ser usado para colar em operadores, funções, blocos funcionais e programas. Primeiramente, é sempre inserido um operador "AND". Isso pode ser convertido por seleção e sobrescrição do texto em outro operador, em outra função, em outro bloco funcional ou outro programa. O Assistente de Entrada serve para selecionar o bloco desejado a partir da lista de blocos suportados. Se o novo bloco tiver outro número mínimo de entradas, essas serão anexadas. Se o novo bloco tem um número mais alto menor de entradas, as últimas entradas serão excluídas.

'Inserir' 'Entrada' no CFC

Símbolo:

Atalho: <Ctrl>+<E>

Este comando é usado para inserir uma entrada. O texto oferecido "??? " pode ser selecionado e substituído por uma variável ou constante. O Assistente de entrada também pode ser usado aqui.

'Inserir' 'Saída' no CFC

Símbolo:

Atalho: <Ctrl>+<A>

Este comando é usado para inserir uma saída. O texto oferecido "??? " pode ser selecionado e substituído por uma variável. O valor que está associado com a entrada da saída é alocado a esta variável.

'Inserir' 'Salto' no CFC

Símbolo:

Atalho: <Ctrl>+<J>

Este comando é usado para inserir um salto. O texto oferecido "??? " pode ser selecionado e substituído pelo rótulo do salto para qual o programa deve saltar.

O rótulo do salto é inserido usando o comando **'Inserir' 'Rótulo'**.

'Inserir' 'Rótulo' no CFC

Símbolo:

Atalho: <Ctrl>+<L>

Este comando é usado para inserir um rótulo. O texto oferecido "??? " pode ser selecionado e substituído pelo rótulo do salto . No modo Online, um rótulo de RETORNO para marcar o final da POU é automaticamente inserido.

O salto é inserido usando o comando **'Inserir' 'Salto'**.

'Inserir' 'Retorno' no CFC

Símbolo:

Atalho: <Ctrl>+<R>

Este comando é usado para inserir um comando de RETORNO. Observe que no modo Online um rótulo de salto com o nome RETURN é automaticamente inserido na primeira coluna e após o último elemento no editor; nas etapas, ele é automaticamente pulado antes que a execução saia da POU.

'Inserir' 'Comentário' no CFC

Símbolo:

Atalho:<Ctrl>+<K>

Este comando é usado para inserir um comentário.

Obtém-se uma nova linha dentro do comentário com<Ctrl> + <Enter>.

'Inserir' 'Entrada de Caixa no CFC

Atalho:<Ctrl>+<U>

Este comando é usado para inserir uma entrada em uma caixa. O número de entradas varia para muitos operadores (ex. ADD pode ter duas ou mais entradas).

Para aumentar o número de entradas em um para cada operador, a caixa propriamente dita deve estar marcada.

'Inserir' 'Pino de Entrada' no CFC, 'Inserir' 'Pino de Saída'

Símbolo:

Esses comandos estão disponíveis, assim que uma macro for aberta para edição. Eles são usados para inserir pinos de entrada e de saídas como nas entradas e saídas da macro. Eles diferem das entradas e saídas normais de POUs pela forma como são exibidos e por não terem nenhum índice de posição.

'Extras' 'Negar' no CFC

Símbolo:

Atalho:<Ctrl>+<N>

Esse comando é usado para negar as entradas, saídas, saltos ou comandos de RETORNO. O símbolo para a negação é uma pequena cruz na conexão.

A entrada do bloco de elemento, saída, salto ou retorno é negada quando ela está selecionada.

A saída do bloco de elemento ou entrada é negada quando ela estiver selecionada (cursor posição 4).

Uma negação pode ser excluída negando-a novamente.

'Extras' 'Set/Reset' no CFC

Símbolo:

Atalho:<Ctrl>+<T>

Este comando só pode ser usado para entradas selecionadas da saída do elemento.

O símbolo para Set é S e para Reset é R.



Figura 5-43. 'Set/Reset' no CFC

VarOut1 é definida como TRUE, se VarIn1 transmite TRUE. VarOut1 retém esse valor, mesmo quando VarIn1 retorna como FALSE.

VarOut2 é definida como FALSE, se VarIn2 oferece TRUE. VarOut2 retém esse valor, mesmo quando VarIn2 retorna como FALSE.

A ativação múltipla desse comando faz com que a saída alterne entre liga, desliga e a condição normal.

'Extras' 'EN/ENO' no CFC

Símbolo:

Atalho: <Ctrl>+<I>

Este comando é usado para dar a um bloco selecionado (posição do cursor 3), uma entrada de habilitação Booleana adicional EN (Habilitar entrada) e uma saída Booleana ENO (Habilitar saída).

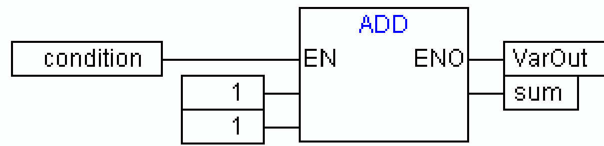


Figura 5-44. 'EN/ENO' no CFC

ADD só é executada neste exemplo quando a variável booleana "condition" é TRUE. VarOut também será definido como TRUE após a execução de ADD. Mas, se posteriormente a condição for alterado para FALSE, ADD não será mais executada e, portanto, VarOut permanece TRUE! O exemplo a seguir mostra como o valor ENO pode ser usado para mais blocos:

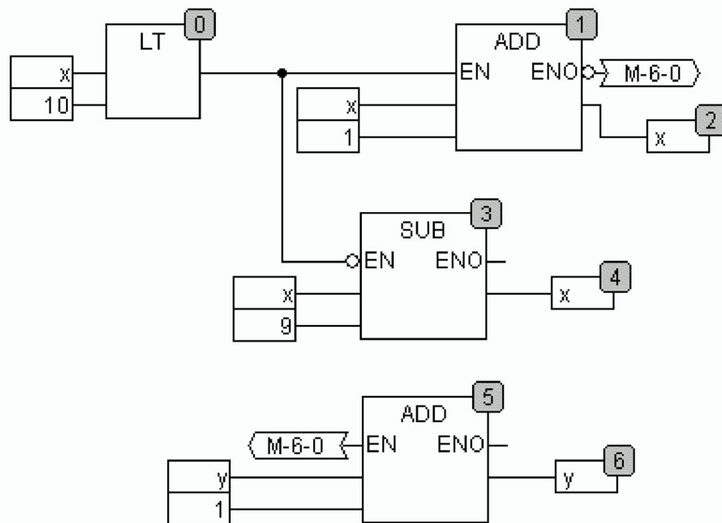


Figura 5-45. ENO usado para mais blocos

x deve inicializado com 1 e y inicializado com 0. Os números no canto direito do bloco indicam a ordem na qual os comandos são executados.

x será aumentado em um até que ele atinja o valor 10. Isso faz com que a saída do bloco LT (0) transmita o valor FALSE e SUB (3) e ADD (5) sejam executadas. x é definido novamente com o valor 1 e y é incrementado em 1. LT (0) é executado novamente desde que x seja menor do que 10. y, portanto, conta o número de vezes que x passa pelo intervalo de valores de 1 a 10.

'Extras' 'Propriedades… ' no CFC

Parâmetros de entrada do tipo constante (VAR_INPUT CONSTANT) de funções e blocos funcionais não são mostrados diretamente no Editor CFC. Eles podem ser exibidos e seus valores podem ser alterados ao selecionar o tronco do bloco em questão e, em seguida, selecionando o comando 'Extras' 'Propriedades' ou simplesmente clicando duas vezes no tronco. O diálogo "Editar parâmetros" é aberto.

Caixa de diálogo de propriedades:

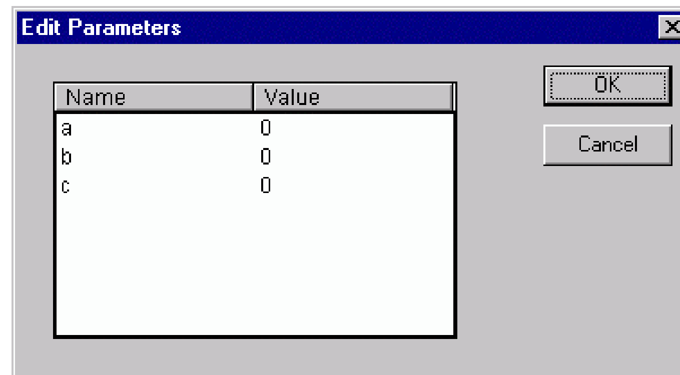


Figura 5-46. Diálogo de propriedades no CFC

Os valores do parâmetro de entrada constante (VAR_INPUT CONSTANT) podem ser alterados. Aqui é necessário marcar o valor de parâmetro na coluna Valor. Outro clique do mouse ou pressionando a barra de espaço permite que sejam editados. A confirmação da alteração do valor é feita pressionando a tecla <Enter>. Pressionando <Escape> as alterações são rejeitadas. O botão OK armazena todas as alterações que foram feitas.

Considerar: essa funcionalidade e a declaração associada de variáveis com a palavra-chave "VAR_INPUT CONSTANT" é somente de impacto no Editor CFC. No Editor FBD todas as variáveis de entrada sempre serão exibidas em uma caixa, não importa se declaradas como VAR_INPUT ou VAR_INPUT CONSTANT. Também para editores de texto isso não faz qualquer diferença.

Selecionando elementos no CFC

Clicar no tronco do elemento para selecioná-lo.

Para marcar mais elementos pressionar a tecla <Shift> e clicar nos elementos necessários, um após o outro, ou arrastar o mouse (esquerda), mantendo o mouse pressionado sobre os elementos a serem marcados.

O comando **'Extras' 'Selecionar todos'** marca todos os elementos de uma vez.

Movendo elementos no CFC

Um ou mais elementos selecionados podem ser movidos com as teclas de direção enquanto a tecla <Shift> é pressionada. Outra possibilidade é mover os elementos usando a tecla esquerda do mouse pressionada. Esses elementos são colocados, liberando a tecla esquerda do mouse, aonde não conflitem ou excedam o tamanho previsto do editor. O elemento marcado salta de volta para sua posição inicial em tais casos e um som de advertência apita.

Copiando elementos no CFC

Um ou mais elementos selecionados podem ser copiados com o comando **'Editar' 'Copiar'** e inseridos com o comando **'Editar' 'Colar'**.

Criando conexões

Uma entrada de um elemento pode ser conectada com precisão à saída de outro elemento. Uma saída de um elemento pode ser conectada às entradas de um número de outros elementos.

Há um número de possibilidades para se conectar a entrada de um elemento E2 com a saída de um elemento E1.



Figura 5-47. Conexões no CFC

Coloque o mouse na saída do elemento E1, clique com a tecla esquerda do mouse, segure-a e arraste o cursor do mouse para a entrada do elemento E2, liberando a tecla esquerda do mouse. Uma conexão é feita a partir da saída de Elemento E1 para o cursor do mouse durante esta operação de arrastar com o mouse.

Coloque o mouse na entrada do elemento E2, clique com a tecla esquerda do mouse, segure-a e arraste o cursor do mouse para a entrada do elemento E1, liberando a tecla esquerda do mouse.

Mova um dos elementos E1 ou E2 e coloque-os de tal forma, que a saída do elemento E2 e a entrada de elemento E1 se encontrem. Faça isto, soltando a tecla esquerda do mouse.

Se o elemento E2 é um bloco com uma entrada livre, uma conexão pode também ser feita arrastando o mouse a parir de uma saída de E1 para o tronco de E2. Uma conexão com a entrada livre na posição mais alta no E2 será criada quando a tecla do mouse é liberada. Se o bloco E2 não tem uma entrada livre, mas é um operador que pode ter uma entrada adicionada a ela, uma nova entrada será automaticamente gerada.

A saída e entrada de um bloco podem ser conectadas juntas (caminho de realimentação) usando esse método. Para estabelecer uma conexão entre dois pinos, clique com o botão esquerdo do mouse em um pino, mantenha o botão pressionado e arraste a conexão para o pino desejado, e solte o botão. Se durante o arrastamento a conexão estender-se fora da área de trabalho do editor, a rolagem ocorre automaticamente. Para tipos de dados simples, o tipo de teste é executado durante a conexão. Se o tipo dos dois pinos não for compatível, o cursor se altera para “Proibido”. Para tipos de dados complexos, nenhum teste ocorre.

Mudando conexões

Uma conexão entre a saída de um elemento E1 e a entrada de um elemento E2 facilmente pode ser transformada em uma conexão entre a saída de Elemento E1 e a entrada de elemento E3. Um clique do mouse é realizado na entrada de E2, a tecla esquerda do mouse é mantida pressionada, o cursor do mouse é movido para a entrada de E3 e, então, é liberada.

Excluindo conexões

Há um número de possibilidades para remover a conexão entre a saída de um elemento E1 e a entrada de um elemento E2:

Selecione a saída do Elemento E1 e pressione a tecla <Delete> ou execute o comando **'Editar' 'Excluir'**. Várias conexões serão removidas mesmo se a saída de E1 estiver conectada a mais de uma das entradas.

Selecione a entrada do elemento E2 e pressione a tecla <Delete> ou execute o comando **'Editar' 'Excluir'.**

Selecione a entrada de E2 com o mouse, mantenha a tecla esquerda do mouse pressionada e arraste a conexão a parir da entrada para fora de E2. A conexão é removida quando a tecla esquerda do mouse é solta em uma área livre da tela.

'Extras' ' Marcador de conexão '

Conexões também podem ser representadas por um conector (marcador de conexão) em vez de uma linha de conexão. Aqui a saída e a entrada associada têm um conector adicionado a elas, ao qual é fornecido um nome exclusivo.

Onde já existe uma conexão entre dois elementos, que agora deve ser representada por conectores, a saída da linha de conexão é marcada e o ponto de menu ‘Extras' ‘Marcador de conexão' é selecionado. O diagrama a seguir mostra uma conexão antes e após a seleção desse ponto no menu.

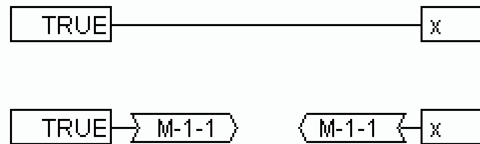


Figura 5-48. Conexão antes/após (CFC)

Um nome exclusivo é fornecido como padrão pelo programa que começa com M, mas que pode ser alterado. O nome do conector é armazenado como um parâmetro de saída, mas pode ser editado na entrada e na saída.

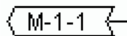
É importante saber que o nome do conector é associado a uma propriedade da saída de uma conexão e é armazenado com ela.

1. Edite o conector na saída:



Se o texto no conector é substituído, o novo nome do conector é adotado por todos os conectores associados nas entradas. No entanto, não é possível selecionar um nome que já pertence a outro marcador de conexão, uma vez que a exclusividade do nome do conector seria violada.

2. Edite o conector na entrada:



Se o texto em um conector é substituído, ele também será substituído no marcador de conexão correspondente na outra POU. Conexões em representações de conectores podem ser convertidas em conexões normais onde se marcam a saída das conexões (posição do cursor 4) e novamente seleciona-se o ponto de menu **'Extras' 'Marcador de conexão'**.

Inserir entradas/saídas flutuantes

Se exatamente um pino de entrada ou saída de um elemento é selecionado, então o elemento de E/S correspondente pode ser diretamente inserido e seu campo de edição preenchido com uma string via teclado.

Ordem de execução

O bloco de elementos de saída, salto, retorno e rótulo possuem um número que indica a ordem na qual são executados. Nessa ordem seqüencial os elementos individuais são avaliados no controlador.

Ao colar em um elemento o número é automaticamente fornecido de acordo com a seqüência topológica (da esquerda para direita e de cima para abaixo). O novo elemento recebe o número de seu sucessor topológico se a seqüência já foi alterada e todos os números mais altos foram aumentados em um.

O número de um elemento permanece constante quando ele é movido.

A seqüência influencia o resultado e deve ser alterada em certos casos.

Se a seqüência for exibida, o número de execução seqüencial correspondente é mostrado no canto direito do elemento.

'Extras' 'Ordenar' 'Mostrar Ordem'

Este comando alterna a exibição da ordem de execução. A configuração padrão é para exibi-la (reconhecido por um sinal (✓) na frente do ponto do menu).

A ordem relevante do número de execução aparece no canto superior direito para o bloco de elementos, saída, salto, retorno e rótulo.

'Extras' 'Ordenar' 'Ordenar topologicamente'

Elementos são classificados em uma seqüência topológica quando a execução ocorre da esquerda para direita e de cima para baixo (incremento do número) para elementos topologicamente organizados. As ligações não são relevantes, apenas o local dos elementos é importante.

Todos os elementos selecionados são organizados topologicamente quando o comando 'Extras' 'Ordem' 'Ordenar topologicamente' é executado. Todos os elementos na seleção são tomados a partir da lista de processamento seqüencial remanescente por este processo. Os elementos dão entrada na lista de processamento seqüencial remanescente, individualmente, da direita inferior até a parte superior esquerda. Cada elemento marcado é inserido na lista seqüencial de processamento antes de seu sucessor topológico, ou seja, ele é inserido antes do elemento que, em um seqüenciamento topológico, iria ser executado depois, quando todos os elementos no editor foram seqüenciados acordo com um sistema de seqüenciamento topológico. Isso será analisado através de um exemplo.

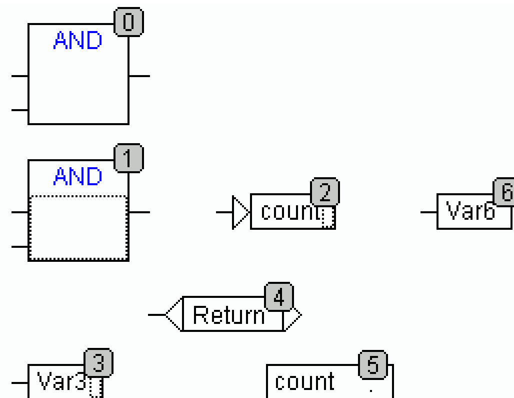


Figura 5-49. Exemplo de ordenação topológica

Os elementos com números 1, 2 e 3 estão selecionados. Se o comando 'Ordenar topologicamente' é selecionado os elementos são primeiro retirados da lista de processamento seqüencial. Var3, o salto e o operador AND são, então, inseridos novamente um após o outro. Var3 é colocado antes do rótulo e recebe o número 2. O salto, em seguida, é ordenado e recebe o número 4 primeiramente, porém torna-se 5 depois que o AND é inserido. A nova ordem de execução que se apresenta é:

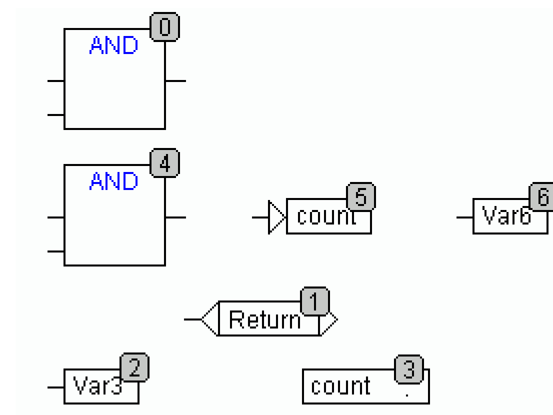


Figura 5-50. Nova ordem de execução em função da ordenação topológica

Quando é introduzido um bloco novo, o mesmo será colocado por padrão na frente do seu sucessor topológico na lista seqüencial de processamento.

'Extras' 'Ordenar' 'Um acima'

Com esse comando todos os elementos selecionados com exceção do elemento que está no início da lista de processamento seqüencial são movidos para um lugar à frente na lista de processamento seqüencial.

'Extras' 'Ordenar' 'Um abaixo'

Com esse comando todos os elementos selecionados com exceção do elemento que está no final da lista de processamento seqüencial são movidos para um lugar atrás na lista de processamento seqüencial.

'Extras' 'Ordenar' 'Início'

Com esse comando todos os elementos selecionados serão movidos para a frente da lista de processamento seqüencial na qual a ordem dentro do grupo de elementos selecionados é mantida. A ordem dentro do grupo de elementos desmarcados também permanece a mesma.

'Extras' 'Ordenar' 'Fim'

Com esse comando todos os elementos selecionados serão movidos para o final da lista de processamento seqüencial na qual a ordem dentro do grupo de elementos selecionados é mantida. A ordem dentro do grupo de elementos desmarcados também permanece a mesma.

'Extras' 'Ordenar' 'Ordenar tudo de acordo com o fluxo de dados'

Esse comando afeta **todos** os elementos. A ordem de execução é determinada pelo fluxo de dados dos elementos e não por sua posição.

O diagrama a seguir mostra elementos que foram ordenados topograficamente.

Seqüência antes da ordenação de acordo com o fluxo de dados:

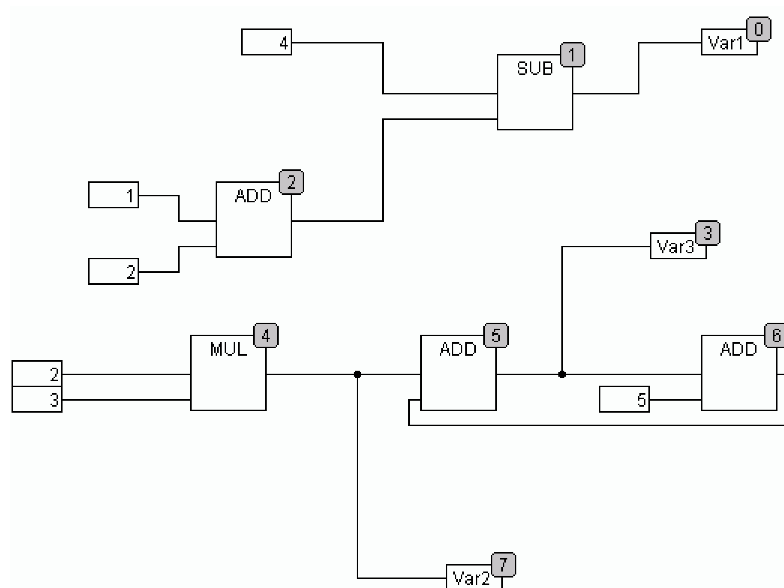


Figura 5-51. Elementos ordenados topograficamente no CFC

A seguinte organização existe depois de selecionar o comando (seqüência após a ordenação de acordo com o fluxo de dados):

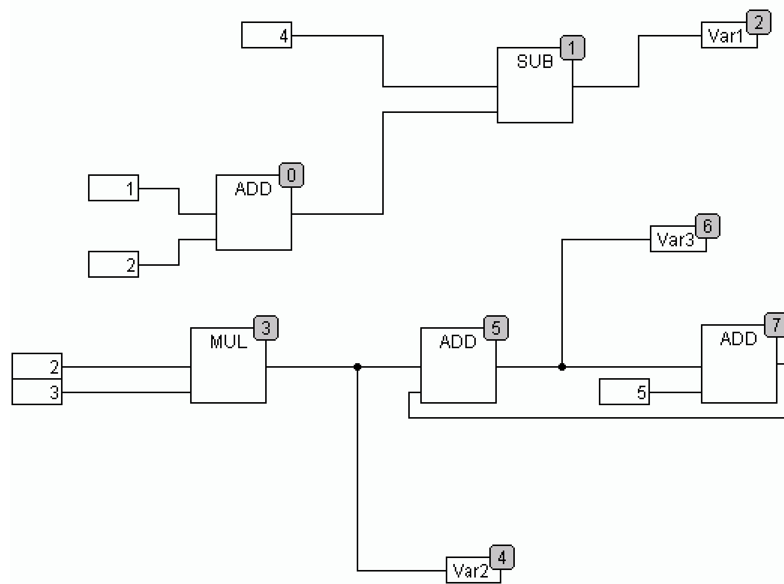


Figura 5-52. Ordenação de acordo com o fluxo de dados

Quando esse comando é selecionado, a primeira coisa a acontecer é que os elementos são classificados topograficamente. Uma nova lista de processamento seqüencial, então, é criada. O computador, com base nos valores conhecidos das entradas, calcula quais dos elementos numerados ainda não numerados podem ser processados em seguida. Na "rede" acima, o bloco AND, por exemplo, pode ser processado imediatamente desde que os valores em suas entradas (1 e 2) sejam conhecidos. O bloco SUB somente então pode ser processado, desde que o resultado de ADD seja primeiro conhecido, etc..

Caminhos de realimentação são inseridos por último.

A vantagem do seqüenciamento de fluxo de dados é que uma caixa de saída que está conectada à saída de um bloco vem imediatamente após o mesmo. Em um sistema de seqüenciamento de fluxo de dados que não esteja topologicamente ordenado, este não seria sempre o caso. A ordenação

topológica pode fornecer outro resultado em alguns casos não ordenados pelo fluxo de dados, um ponto que pode-se reconhecer no exemplo acima.

'Extras' 'Criar Macro'

Símbolo:

Com esse comando, várias POUs selecionadas ao mesmo tempo podem ser montadas em um bloco, que pode ser nomeado como uma macro. As macros só podem ser reproduzidas por Copiar/Colar, na qual cada cópia torna-se uma macro separada cujo nome pode ser escolhido independentemente. As macros, portanto, não são as referências. Todas as conexões que são cortadas pela criação de uma macro geram pinos de entrada ou saída na macro. Conexões com entradas geram um In-Pin (pino de entrada). O nome padrão aparece próximo ao pino de entrada sob a forma In<n>. Para conexões para saídas, Out<n> aparece. Conexões afetadas que tinham marcadores de conexão antes da criação da macro, mantêm o marcador de conexão com o PINO da macro.

Primeiramente, uma macro tem o nome padrão "MACRO". Isso pode ser alterado no campo Nome da macro em uso. Se a macro é editada, seu nome será exibido na barra de título da janela do editor acrescentada ao nome da POU.

Exemplo Seleção:

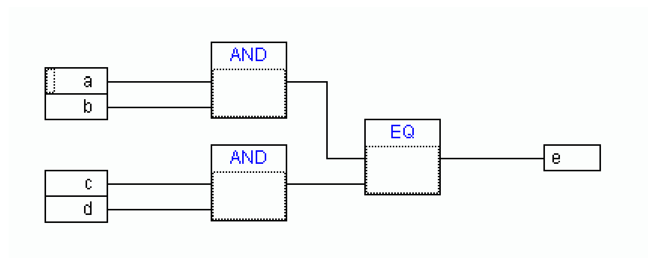


Figura 5-53. Macro (Exemplo Seleção)

Macro:

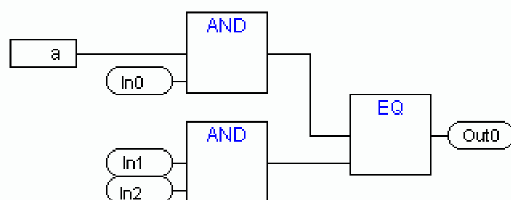


Figura 5-54. Macro

No editor:

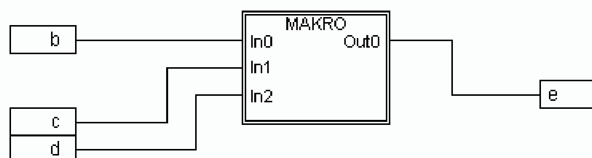


Figura 5-55. Macro (no editor)

'Extras' 'Editar Macro'

Símbolo:

Por esse comando ou clicando duas vezes no corpo da macro, ela é aberta para edição na janela do Editor da POU associada. O nome da macro é exibido acrescentado ao nome da POU na barra de título.

As caixas de pino geradas para as entradas e saídas da macro durante a criação podem ser manipuladas como entradas e saídas normais de POU. Elas podem também ser movidas, excluídas, adicionadas, etc.. Elas diferem apenas em como são exibidas (não tem nenhum índice de posição).

Para adicionar pode-se usar os botões de (entrada) e (saída), respectivamente, que estão disponíveis na barra de menu. Caixas de PINOS têm cantos arredondados. O texto na caixa de pino corresponde ao nome do pino na exibição da macro.

A ordem dos pinos na caixa macro segue a ordem da execução dos elementos da macro. Um índice de ordem inferior antes de um maior e pino maior antes de um mais baixo.

A ordem de processamento dentro a macro é encerrada, em outras palavras a macro é processada como um bloco, na posição da macro na POU primária. Comandos para manipular a ordem da execução, portanto, operam somente dentro da macro.

Extras' 'Expandir Macro'

Com esse comando, a macro selecionada é expandida novamente e os elementos nela contidos são inseridos na POU no local da macro. As conexões com os pinos da macro novamente são exibidas como conexões de entrada e saídas dos elementos. Se a expansão da macro não pode ocorrer no local da caixa da macro por falta de espaço, a macro é movida para a direita e para baixo até que espaço esteja disponível.

Nota: se o projeto é salvo sob a versão de número 2.1, as macros serão todas expandidas da mesma forma. Todas as macros também serão expandidas antes da conversão em outras linguagens.

'Extras' 'Uma macro nível anterior', 'Extras' 'Todas macros nível anterior'

Símbolos:

Esses comandos também estão disponíveis na barra de ferramentas, assim que uma macro é aberta para edição. Se as macros estão aninhadas dentro de outra, é possível alternar para o próximo maior ou para o nível mais alto de exibição.

Caminhos de realimentação no CFC

Caminhos de realimentação só podem ser exibidos diretamente no editor de diagrama contínuo de funções e não no editor de diagrama de blocos funcionais usuais. Aqui deve ser observado que a saída de um bloco sempre executa uma variável intermediária interna. O tipo de dados da variável intermediária resulta, para operadores, a partir do maior tipo de dados das entradas.

O tipo de dados de uma constante é obtido a partir o menor tipo de dados possíveis, isto é a constante '1' adota o tipo de dados SINT. Se agora uma adição com realimentação e a constante '1' é executada, a primeira entrada oferece o tipo de dados SINT e o segundo é indefinido devido à realimentação. Assim, a variável intermediária também é do tipo SINT. O valor da variável intermediária só então é alocado à variável de saída.

O diagrama abaixo mostra uma adição com realimentação e uma adição com uma variável. As variáveis x e y devem ser do tipo INT.

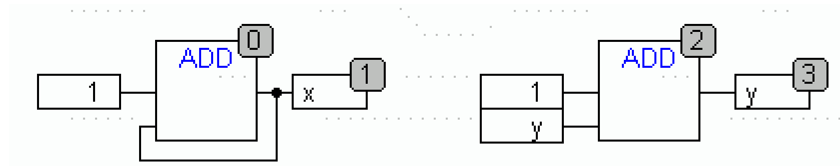


Figura 5-56. Adição com realimentação e adição com uma variável no CFC

Há diferenças entre as duas adições...

A variável y pode ser inicializada com um valor que não é igual a zero, mas isso não é o caso de variável intermediária para a adição à esquerda.

A variável intermediária para a adição à esquerda é do tipo SINT enquanto que à direita é do tipo de dados INT. As variáveis x e y têm valores diferentes após a chamada de número 129. A variável x, embora seja do tipo INT, contém o valor 127 porque a variável intermediária foi para overflow. A variável y, por outro lado, contém o valor 129.

CFC no modo Online

Monitoração:

Os valores para as entradas e saídas são exibidos dentro de caixas de entrada ou saídas. Constantes não são monitoradas. Para variáveis não-booleanas, as caixas são expandidas para acomodar os valores exibidos. Para conexões booleanas, o nome da variável, bem como a conexão será exibida em azul se o valor é TRUE, caso contrário, elas permanecem em preto.

Conexões booleanas internas também são exibidas online em azul no estado TRUE, caso contrário, preto. O valor de conexões não-booleanas internas é exibido em uma caixa pequena com cantos arredondados no pino de saída da conexão.

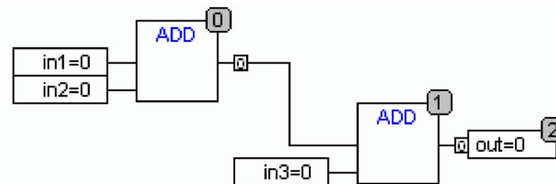


Figura 5-57. Monitoração em CFC

Os pinos nas macros são monitorados como em nas caixas de entrada ou de saída.

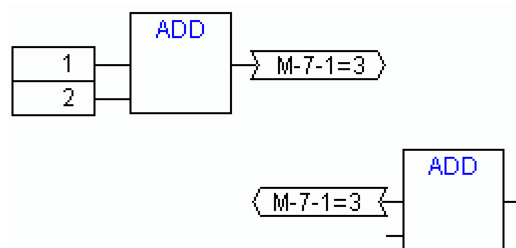


Figura 5-58. Monitoração das macros em CFC

Conexões não-booleanas com marcadores de conexão exibem seu valor dentro do marcador de conexão. Para conexões booleanas, as linhas, bem como os nomes dos marcadores são exibidas em azul se a linha estiver carregando o valor TRUE, caso contrário, preto.

Controle de fluxo:

Quando o controle de fluxo é ligado, as conexões que tenham sido partilhadas são marcadas com a cor selecionada nas opções do projeto.

Breakpoints:

Breakpoints podem ser definidos em todos os elementos que também tenham um índice de ordem de seqüência de processamento. O processamento do programa será interrompido antes da execução do respectivo elemento. Aplicação: POUs e saídas antes da atribuição de entradas, bem como rótulos de salto antes da execução do elemento com o próximo índice. O índice de seqüência de processamento do elemento é usado como a posição do breakpoint na caixa de diálogo do breakpoint.

A configuração de breakpoints em um elemento selecionado é realizada com a tecla F9 ou via o item de menu 'Lig./Desl'. Breakpoint' no menu ‘Comunicação’ ou 'Extras' ou no menu de contexto do editor. Se um breakpoint for definido em um elemento, então, este será apagado e revertido na próxima vez que o comando 'Lig./Desl'. Breakpoint ' for executado. Além disso, o breakpoint em um elemento pode ser alternado clicando duas vezes nele.

Breakpoints são exibidos em cores inseridas nas opções de projeto.

Rótulo de RETORNO:

No modo online, um rótulo de salto com o nome „RETURN" é gerado automaticamente na primeira coluna e após o último elemento no editor. Este rótulo marca o fim da POU (saltar para) quando a etapa anterior à execução sai da POU. Marcas NO RETURN são inseridas em macros.

Passos:

Quando usando 'Sobrepasso', o elemento com o índice de ordem superior mais próximo sempre será saltado. Se o elemento atual for uma macro ou uma POU, então sua implementação ramifica quando 'Passo a passo' está em vigor. Se um 'Sobrepasso' for executado a partir daí, o elemento cujo índice de ordem segue o da macro é saltado.

Zoom para a POU

Atalho: <Alt>+<Enter>

Com este comando uma POU selecionada é carregada em seu editor. O comando está disponível no menu de contexto (<F2>) ou no menu 'Extras', se o cursor estiver posicionado sobre o nome de uma POU em um editor de texto ou se a caixa POU é selecionada em um editor gráfico.

Ao lidar com uma POU de uma biblioteca, então, o gerenciador de bibliotecas é chamado e a POU correspondente será exibida.

6. Recursos

Visão geral dos recursos

No cartão de registro **Recursos** do Organizador de Objetos, existem objetos para configurar e organizar o seu projeto e para rastrear os valores das variáveis:

Variáveis globais que podem ser utilizadas no projeto inteiro; as variáveis globais do projeto, bem como as bibliotecas.

Gerenciador de Bibliotecas para tratamento de todas as bibliotecas que estão incluídas ao projeto

Configuração do CP para configurar o hardware

Gerenciador de Monitoração e Receitas para indicar e predefinir valores das variáveis

Configuração de tarefas para controlar seu programa de controle através de tarefas

Configurações do dispositivo para selecionar a plataforma de hardware (dispositivo) e, se disponível, para customizar parâmetros específicos do dispositivo

Área de trabalho como uma imagem das opções do projeto

Dependendo das configurações do dispositivo os seguintes recursos também podem estar disponíveis:

Gerenciador de parâmetros para gerir variáveis, que também estão acessíveis para outros participantes na rede. Esta funcionalidade só estará disponível se definido nas correspondentes configurações do dispositivo.

Navegador CP para monitoramento das informações provenientes do CP

Ferramentas para conectar ferramentas externas, que, então, podem ser iniciadas no MasterTool IEC. Esta funcionalidade só estará disponível se definido nas correspondentes configurações do dispositivo.

Além disso, pode ser criado e carregado um **arquivo de documentação**, o qual oferece um conjunto de comentários para as variáveis de projeto (por exemplo, em certo idioma) e que será impresso quando documentando o projeto via 'Projeto', 'Documentar'.

Guia Recursos no MasterTool IEC:

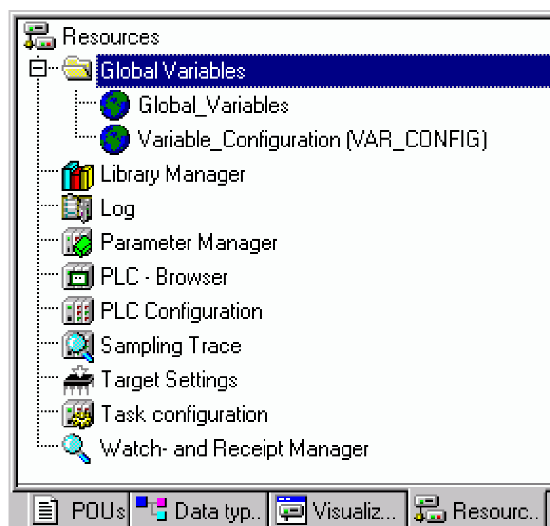


Figura 6-1. Guia Recursos no MasterTool IEC

Variáveis globais e configuração de variáveis

Variáveis globais

Objetos em 'Variáveis globais'

No Organizador de Objetos, encontram-se dois objetos no cartão de registro **Recursos**, na pasta de **Variáveis Globais** (nomes padrão dos objetos em parênteses).

Lista de Variáveis Globais (Variáveis Globais)

Configuração de Variáveis (Configuração de Variáveis)

Todas as variáveis definidas nesses objetos são reconhecidas ao longo do projeto.

Se a pasta Variáveis Globais não estiver aberta (sinal de adição na frente da pasta), pode-se abri-la com um clique duplo <Enter> na linha.

Selecione o objeto correspondente. O comando **'Abrir objeto'** abre uma janela com as variáveis globais definidas anteriormente. Para isso, o editor funciona exatamente como o editor de declaração.

Várias listas de variáveis

Variáveis globais, variáveis de rede globais (**VAR_GLOBAL**), variáveis de rede globais (**VAR_GLOBAL**, específico do dispositivo) e configurações de variáveis (**VAR_CONFIG**) deverão ser definidas em objetos separados.

Se foi declarado um grande número de **variáveis globais** e quer estruturar melhor sua lista de variáveis globais, então, pode-se criar outras listas de variáveis.

No Organizador de objetos, selecione a pasta **Variáveis Globais** ou um dos objetos existentes com variáveis globais. Em seguida, execute o comando **'Projeto' 'Acrescentar objeto'**. Dê um nome correspondente ao objeto que aparece na caixa de diálogo. Com esse nome, um objeto adicional será criado com a palavra-chave **VAR_GLOBAL**. Para um objeto com uma configuração de variável, altere a correspondente palavra-chave para **VAR_CONFIG**.

O que são Variáveis Globais

Variáveis “Normais”, constantes ou variáveis remanentes utilizadas ao longo do projeto podem ser declaradas como variáveis globais, mas também como variáveis de rede que também são usadas para a troca de dados com outros assinantes da rede.

Considerar: em um projeto, pode-se definir uma variável local que tenha o mesmo nome de uma variável global. Nesse caso dentro de uma POU será usada a variável localmente definida.

Não é permitido nomear duas variáveis globais de forma idêntica. Por exemplo, ocorrerá um erro de compilador, se tiver definido uma variável "var1" na configuração do CP e também em uma lista de variáveis globais.

Variáveis de Rede

Nota: o uso de variáveis de rede deve ser suportado pelo sistema do dispositivo e deve ser ativado nas configurações do dispositivo (categoria funcionalidade de rede).

Através de uma troca de dados automática (compare isso com a troca de dados não-automática via Gerenciador de parâmetro) é possível atualizar o valor de uma variável de rede em vários sistemas de controladores dentro de uma rede do controlador compatível no **MasterTool IEC**. Isso não exige funções específicas de controlador, mas os assinantes de rede devem usar listas de declaração idênticas e configurações de transferência correspondentes para variáveis de rede em seus projetos. Para tornar isso possível, é recomendável que a declaração não seja inserida manualmente em cada aplicativo do controlador, mas carregada a partir de um arquivo separado quando criar a lista. (Consultar 'Criar uma lista de variáveis globais').

Para a troca de variáveis de rede é necessário que as variáveis sejam usadas em uma tarefa cíclica ou tarefa de execução livre ou no PLC_PRG. Não é suficiente declará-las na parte de declarações da POU! Se as variáveis forem usadas em várias tarefas/PLC_PRG, então, na troca de dados, essas tarefas serão consideradas aquelas de prioridade mais alta.

Criar uma Lista de Variáveis Globais

Para criar uma lista de variáveis globais, abra o registro 'Recursos' no Organizador de Objetos e selecione a entrada 'Variáveis globais' ou selecione uma lista já existente. Então, escolha o comando 'Projeto' 'Objeto' 'Acrescentar' para abrir a caixa de diálogo de lista de variáveis globais.

Esta caixa de diálogo também pode ser aberta pelo comando 'Projeto' 'Objeto' 'Propriedades' que está disponível se uma lista de variáveis globais existente é marcada no organizador de objetos. Ela mostra a configuração desta lista.

Caixa de diálogo para criar uma nova lista de variáveis globais:

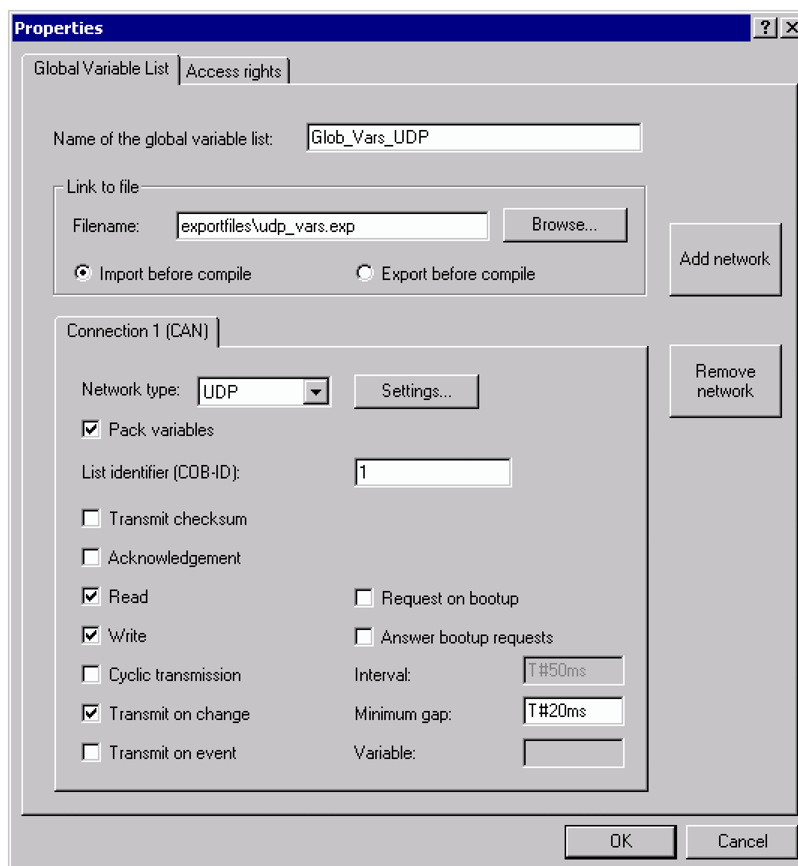


Figura 6-2. Caixa de diálogo para criar uma nova lista de variáveis globais

Nome da lista de variáveis globais: inserir um nome para a lista.

Vincular ao Arquivo:

Nome do arquivo: no caso de um arquivo de exportação (* .exp) ou um arquivo DCF, que contenham as variáveis desejadas, pode-se configurar um link para esse arquivo. Para fazer isso, insira o caminho do arquivo no campo **Nome do arquivo** e pressione o botão "Procurar" para obter a caixa de diálogo padrão 'Selecionar arquivo de texto'. Arquivos DCF são convertidos para sintaxe ICE quando são lidos.

Ative a opção **Importar antes de compilar** para que a lista de variáveis seja lida a partir do arquivo externo antes de cada compilação do projeto. Ative a opção de **Exportar antes de compilar**, para que a lista de variáveis seja gravada no arquivo externo antes de cada compilação do projeto.

Ao fechar a caixa de diálogo 'Lista de Variáveis Globais' com **OK**, o novo objeto é criado. A lista de variáveis globais pode ser reconhecida no Organizador de objetos pelo símbolo apropriado. Com o comando 'Projeto' 'Objeto' 'Propriedades' pode-se reabrir a caixa de diálogo de configuração 'Lista de variáveis globais' para a entrada marcada no Organizador de objetos.

Configuração de variáveis de rede:

Se a opção 'Suportar variáveis de rede' estiver ativada nas configurações do dispositivo, então, o botão <Acrescentar rede> estará disponível. Pressionando esse botão a caixa de diálogo é estendida e aparece como mostrado na figura. Se a opção não está ativada, o botão não estará disponível.

Conexão <n> (<tipo de rede>): na parte inferior da caixa de diálogo é possível criar conjuntos de configuração para até quatro conexões de rede, cada um em uma guia separada. Um conjunto de configuração define os parâmetros da troca de dados para a lista de variáveis específicas dentro da rede. Para a troca na rede funcionar como esperado, a mesma lista de variável deve ser configurada para corresponder aos outros assinantes de rede.

Se nenhuma configuração ainda estiver presente, será exibida no caso de uma rede UDP uma única folha de tabulador com a inscrição **'Conexão 1 (UDP)'**. Cada vez que o botão 'Acrescentar rede' for pressionado novamente, obtém-se até mais quatro folhas inscritas com números de série após “Conexão”.

Tipo de rede: escolha o tipo desejado na lista. A lista é definida por entradas no sistema do dispositivo. Por exemplo, “CAN” como uma abreviação de uma rede CAN ou “UDP” para um sistema de transmissão UDP, podem ser selecionados.

Configurações: esse botão abre a caixa de diálogo de configurações para <tipo de rede> com os seguintes parâmetros de configuração:

UDP:

Usar padrão: se este botão é pressionado, a porta 1202 será definida para a troca de dados com os outros participantes da rede. O endereço de broadcast será configurado como "255. 255. 255. 255", que significa que a troca de dados será feita com todos os participantes na rede.

Porta para todas as redes: digite aqui um número de porta desejada para substituir a configuração padrão (consultar acima, Usar padrão). Certifique-se de que os outros nós na rede estejam definidos na mesma porta! No caso de mais de uma conexão UDP definida no projeto, o número da porta será automaticamente modificado em todos os conjuntos de configuração de acordo com a entrada feita aqui.

Endereço de broadcast: digite aqui um endereço e o intervalo de endereços de uma sub-rede, para o caso de substituir a configuração padrão (por exemplo, "197 . 200 . 100 . 255", para comunicar com todos os nós com endereços IP 197 . 200 . 100 . x).

Considere que para Sistemas Win32, os endereços de broadcast devem corresponder à máscara de sub-rede da configuração TCP/IP do PC!

Diálogo ' configurações para UDP

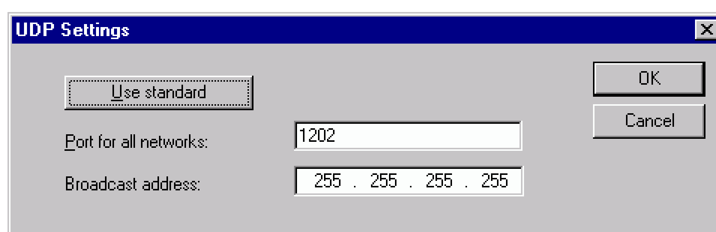


Figura 6-3. Diálogo configurações para UDP

CAN: (consultar disponibilidade)

Índice do controlador: índice do controlador CAN pelo qual as variáveis devem ser transferidas.

As opções a seguir podem ser ativadas ou desativadas na configuração do comportamento de transmissão das variáveis:

Empacotar variáveis: as variáveis são organizadas para a transferência em pacotes (telegramas) cujo tamanho depende da rede. Se a opção estiver desativada, um pacote é configurado para cada variável.

Identificador de lista (COB - id): número de identificação do primeiro pacote, no qual as variáveis serão enviadas. (padrão = 1). Pacotes adicionais serão numerados na ordem ascendente.

Nota: A COB-ID **deve ser exclusiva** dentro de todas as listas de variáveis de rede usadas no projeto. Usar a mesma identificação em várias configurações de lista causará um erro de compilação com as versões do compilador > = 2.3.7.0.

Depende do sistema do dispositivo se as variáveis de rede da lista podem ser definidas como 'leitura' e 'escrita" ou exclusivamente um dos dois. Para definir esta propriedade ativar as respectivas opções 'Ler' e 'Escrever':

Ler: as variáveis na lista serão lidas; se a opção estiver desativada, outras variáveis enviadas pela rede serão ignoradas. A opção a seguir pode ser ativada adicionalmente:

Solicitar na inicialização: se o nó local é um nó de "leitura" (opção 'Ler' ativada), assim que ele for reinicializado, os valores de variáveis atuais serão solicitados a partir de todos os nós de escrita e serão enviados por eles, independentemente de quaisquer outras condições de transmissão (hora, evento), que normalmente disparam a comunicação. Pré-condição: na configuração de nós de escrita a opção 'Responder solicitações de inicialização' deve ser ativada! (consultar abaixo).

Escrever: as variáveis serão escritas, as seguintes opções podem ser definidas adicionalmente:

Incluir verificação: uma verificação será adicionada a cada pacote enviado. A verificação será analisada pelo destinatário para certificar-se de que as definições de variáveis de remetente e destinatário estão idênticas. Um pacote com uma verificação não-correspondente não será aceito e – se isso for configurado ('Usar reconhecimento na transferência', consultar abaixo) – irá ser reconhecida negativamente.

Use reconhecimento de transferência: (nenhuma função no caso de CAN): cada mensagem será confirmada pelo destinatário. Assim que o remetente não obtiver pelo menos uma confirmação dentro de um ciclo, uma mensagem de erro será produzida a qual, no caso de uma rede UDP, será gravada na estrutura de diagnóstico fornecida pelo NetVarUdp_LIB_V23.lib.

Responder solicitações de inicialização: se o nó local é de "escrita" (Opção 'Escrever' ativada), então cada solicitação de um nó de leitura que é enviada por ele na inicialização (solicitação de opção na inicialização, consultar acima), será respondida. Isso significa que os valores das variáveis atuais serão transmitidos mesmo se nenhum outro disparador de transmissão definido (tempo ou evento) estiver forçando isto neste momento.

Transmissão cíclica: variáveis são escritas em intervalos especificados após o Intervalo (ex.: notação de tempo T#70ms).

Transmissão na troca: variáveis são gravadas apenas quando seus valores forem alterados; uma entrada após o mínimo pode, no entanto, definir um lapso de tempo mínimo entre transferências.

Transmitir no evento: as variáveis da lista serão escritas assim que a variável inserida for TRUE.

A lista de variáveis de rede globais é marcada com o símbolo apropriado no Organizador de objetos.

Nota: se uma variável de rede global é usada em uma ou mais **tarefas**, o seguinte aplica-se se para o componente de tempo da transferência: quando cada tarefa é chamada ela é testada para determinar quais parâmetros se aplicam à transferência do valor da variável (configuração na caixa de diálogo 'Lista de variáveis globais'). O valor da variável será transferido ou não, dependendo se o intervalo de tempo especificado tenha passado. Em cada transferência o contador de intervalo de tempo para essa variável é redefinido para zero.

“Enviando” sempre é considerado a partir do controlador afetado. Portanto, funções não específicas de controle têm que ser fornecidas para a troca de dados.

Editando Variáveis Globais e Listas de Variáveis de Rede

O editor para as variáveis globais funciona de maneira semelhante ao editor de declaração. Mas observe que não se pode editar nele uma lista, que é uma imagem de uma lista de variável externa vinculada! Listas de variáveis externas só podem ser editadas externamente e elas serão lidas em cada abertura e compilação do projeto.

Sintaxe:

VAR_GLOBAL

(* declarações de variáveis *)

END_VAR

Variáveis de rede só podem ser usadas, se permitido pelo sistema do dispositivo. Elas também são definidas nessa sintaxe.

Exemplo de uma lista de variáveis de rede que foi criada por meio da vinculação de um arquivo de exportação *.exp com o nome NETWORKVARIABLES_UDP; exemplo de uma lista de variáveis de rede, que foi criada carregando um arquivo de exportação *.exp denominado Network_Vars_UDP:

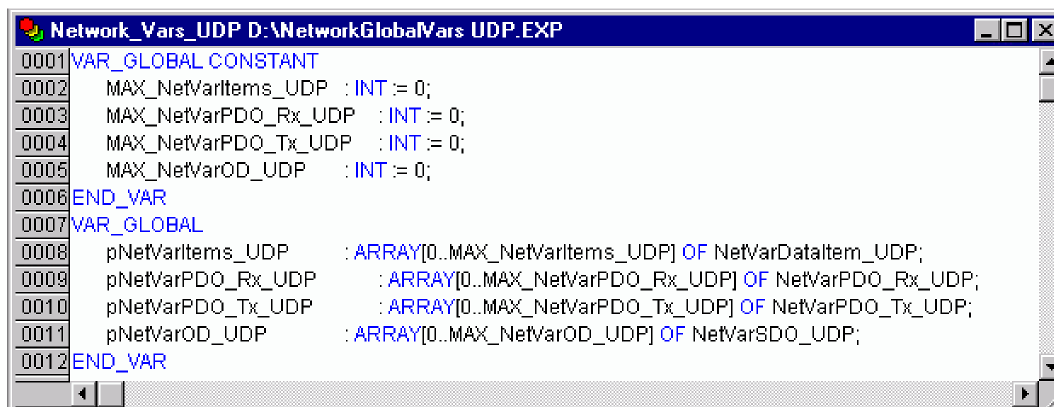


Figura 6-4. Exemplo de uma lista de variáveis de rede via *.exp

Editando listas de Variáveis Globais Remanentes

Se forem suportadas pelo controlador, as variáveis remanentes podem ser processadas. Há dois tipos de variáveis globais remanentes (consultar também o capítulo 5: Variáveis Remanentes)

Variáveis retentivas permanecem inalteradas após um desligamento não controlado do controlador (off/on) ou um ‘Comunicação’ 'Reset' no MasterTool IEC.

Variáveis persistentes permanecem inalteradas apenas após um download do programa.

Variáveis deste tipo não o são automaticamente!

Variáveis Remanentes são adicionalmente atribuídas à palavra-chave **RETAIN** e/ou **PERSISTENT**.

Variáveis de rede também são definidas nessa sintaxe.

Sintaxe:

```
VAR_GLOBAL RETAIN
(* DECLARAÇÕES DE VARIÁVEIS *)
END_VAR
VAR_GLOBAL PERSISTENT
(* DECLARAÇÕES DE VARIÁVEIS *)
END_VAR
```

Para a combinação das propriedades persistentes e retentivas, ambas as palavras-chave são usadas:

VAR_GLOBAL RETAIN PERSISTENT ou VAR_GLOBAL PERSISTENT RETAIN

Variáveis de rede (específico do dispositivo) também são definidas usando esta sintaxe.

Constantes Globais

Constantes globais, adicionalmente têm a palavra-chave **CONSTANT**.

Sintaxe:

```
VAR_GLOBAL CONSTANT
(* DECLARAÇÕES DE VARIÁVEIS *)
END_VAR
```

Configuração de variáveis

Em blocos funcionais é possível especificar endereços para as entradas e saídas que não estão completamente definidas, colocando as definições de variáveis entre as palavras-chave **VAR** e **END_VAR**. Endereços não completamente definidos são identificados com um asterisco.

Exemplo:

```
FUNCTION_BLOCK LOCIO
VAR
 LOCI AT %I*: BOOL := TRUE;
 LOCO AT %Q*: BOOL;
END_VAR
```

Aqui duas variáveis locais de E/S são definidas, uma local de Entrada (%I*) e uma local de Saída (%Q*).

Se desejar configurar E/S locais para configuração de variáveis no Organizador de objetos no cartão de registro de **Recursos**, o objeto **Configuração de Variáveis** geralmente estará disponível. O objeto, então, pode ser renomeado e outros objetos podem ser criados para a configuração de variáveis.

O editor de configuração de variáveis funciona como o editor de declaração.

Variáveis para configurações de E/S locais devem estar localizadas entre as palavras-chave **VAR_CONFIG** e **END_VAR**.

O nome de uma variável consiste em um caminho de instância completa através do qual POUs individuais e nomes de instância são separados uns dos outros por pontos. A declaração deve conter um endereço cuja classe (entrada/saída) corresponde aquela do endereço incompletamente especificado (%I*, %Q*) no bloco funcional. Também o tipo de dados deve concordar com a declaração no bloco funcional.

Variáveis de configuração, cujo caminho de instância é inválido porque a instância não existe, também são indicadas como erros. Por outro lado, um erro também é relatado se nenhuma configuração existir para uma variável de instância. Para receber uma lista de todas as variáveis de configuração necessárias, o item de menu “Todos caminhos de instância” no menu 'Inserir' pode ser usado.

Exemplo para uma configuração de variável (parta do pressuposto de que a seguinte definição para um bloco funcional é fornecida em um programa):

```
PROGRAM PLC_PRG
VAR
HUGO: LOCIO;
OTTO: LOCIO;
END_VAR
```

Então, uma configuração de variável corrigida ficaria assim:

```
VAR_CONFIG
PLC_PRG.HUGO.LOCI AT %IX1.0 : BOOL;
PLC_PRG.HUGO.LOCO AT %QX0.0 : BOOL;
PLC_PRG.OTTO.LOCI AT %IX1.0 : BOOL;
PLC_PRG.OTTO.LOCO AT %QX0.3 : BOOL;
END_VAR
```

'Inserir' 'Todos caminhos de instância'

Com esse comando um bloco **VAR_CONFIG - END_VAR** é gerado contendo todos os caminhos de instância disponíveis no projeto. Declarações já disponíveis não precisam ser reinseridas para conter endereços já existentes. Este item de menu pode ser encontrado na janela para configuração de variáveis se o projeto é compilado ('Projeto' 'Compilar tudo').

Frame de Documento

Se um projeto deve receber várias documentações, talvez com comentários em Alemão e Inglês, ou para documentar vários projetos semelhantes que usem os mesmos nomes de variáveis, então, o trabalho pode ser salvo criando um docuframe com o comando 'Extras' 'Criar arquivo de documentação'.

O arquivo criado pode ser carregado em um editor de texto desejado e pode ser editado. O arquivo começa com a linha **DOCUFILE**. Em seguida, uma lista das variáveis do projeto segue em uma organização que atribui três linhas a cada variável: uma linha **VAR** que mostra quando vem uma nova variável; a seguir, uma linha com o nome da variável; e, finalmente, uma linha vazia. Pode-se agora substituir essa linha, usando um comentário para a variável ou simplesmente excluir quaisquer variáveis que não sejam passíveis de documentação. Se desejado, pode-se criar vários quadros do documento para o projeto.

Editor do Windows com o frame do documento:

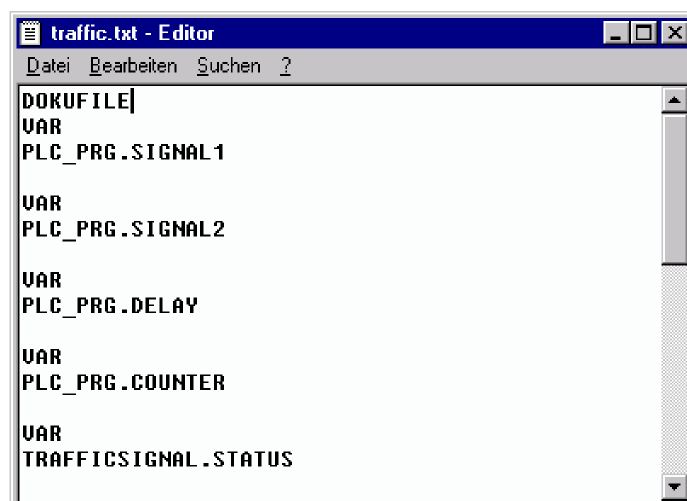


Figura 6-5. Editor do Windows com o frame do documento

Para usar um frame do documento, execute o comando 'Extras' 'Vincular arquivo doc.'. Agora, ao documentar o projeto inteiro, ou imprimir partes do projeto, então, no texto do programa, haverá uma inserção do comentário produzido no docuframe em todas as variáveis. Esse comentário é exibido somente na impressão!

'Extras' 'Criar arquivo de documentação'

Use esse comando para criar um frame do documento. O comando está disponível ao selecionar um objeto a partir das variáveis globais.

Uma caixa de diálogo será aberta para salvar arquivos sob um novo nome. No campo para o **nome do arquivo**, a extensão *.txt já foi inserida. Selecione um nome desejado. Um arquivo de texto é criado então no qual todas as variáveis do seu projeto são listadas.

'Extras' 'Vincular arquivo doc.'

Com esse comando pode-se selecionar um frame do documento.

A caixa de diálogo para abrir arquivos é aberta. Escolha o frame do documento desejado e pressione **OK**. Agora, ao documentar o projeto inteiro, ou imprimir partes do seu projeto, então, no texto do programa haverá uma inserção de um comentário produzido no docuframe em todas as variáveis. Esse comentário é exibido somente na impressão!

Para criar um frame do documento, use o comando 'Extras' 'Criar arquivo de documentação'.

Gerenciador de Bibliotecas

O Gerenciador de Bibliotecas mostra todas as bibliotecas que estão conectadas com o projeto atual. As POUs, tipos de dados e variáveis globais das bibliotecas podem ser usadas da mesma maneira que POUs, tipos de dados e variáveis globais definidos pelo usuário.

Gerenciador de Bibliotecas:

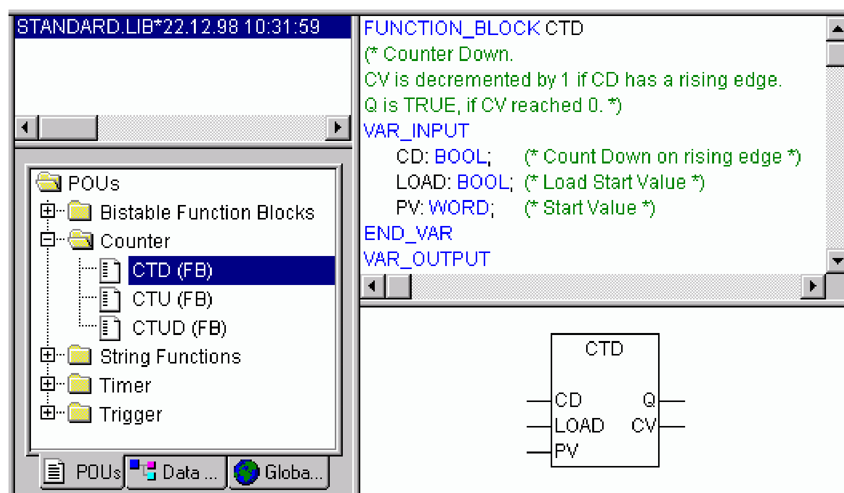


Figura 6-6. Gerenciador de Bibliotecas

O Gerenciador de Bibliotecas é aberto com o comando **'Janela' 'Gerenciador de Bibliotecas'**. As informações relacionadas a bibliotecas incluídas são armazenadas com o projeto e podem ser exibidas em caixa de diálogo 'Informações sobre biblioteca externa'. Para abrir este diálogo selecione o nome da biblioteca correspondente no Gerenciador de Bibliotecas e execute o comando 'Extras' 'Propriedades'.

Usando o Gerenciador de Bibliotecas

A janela do Gerenciador de Bibliotecas é dividida em três ou quatro áreas por divisores de tela. As bibliotecas anexadas ao projeto estão listadas na área superior esquerda.

Na área abaixo desta, dependendo de qual cartão de registro foi selecionado, há uma listagem das **POUs, Tipos de Dados, Visualizações ou Variáveis Globais** da biblioteca selecionados na área superior.

Pastas são abertas e fechadas por cliques duplos na linha ou pressionando <Enter>. Há um sinal de mais na frente das pastas fechadas e um sinal de menos na frente das pastas abertas.

Se uma POU é selecionada, clicando com o mouse ou selecionando com as teclas de setas, então, a declaração da POU é exibida na área superior direita do gerenciador da bibliotecas; e no canto inferior direito é exibido graficamente, na forma de uma caixa preta com as entradas e saídas.

Com tipos de dados e variáveis globais, a declaração é exibida na área à direita do Gerenciador de Bibliotecas.

Biblioteca padrão

A biblioteca com "standard. lib" está sempre disponível. Ela contém todas as funções e blocos funcionais que são solicitados a partir do IEC61131-3, tais como POUs padrão para um sistema de programação IEC. A diferença entre uma função padrão e um operador é que o operador implicitamente é reconhecido pelo sistema de programação, enquanto que POUs padrão devem ser vinculados ao projeto (standard. lib).

O código para essas POUs existe como uma biblioteca-C e é um componente do **MasterTool IEC**.

Bibliotecas definidas pelo usuário

Se um projeto deve ser compilado em sua totalidade e sem erros, então ele pode ser salvo em uma biblioteca com o comando **'Salvar como'** no menu **'Arquivo'**. O projeto em si permanecerá inalterado. Um arquivo adicional com a extensão padrão ".lib" será gerado. Esta biblioteca posteriormente pode ser usada e acessada como, por exemplo, a biblioteca padrão.

Para poder ter os POUs de um projeto disponíveis em outros projetos, salve o projeto como uma **Biblioteca Interna *. lib.** Esta biblioteca depois pode ser inserida em outros projetos usando o Gerenciador de Bibliotecas.

Dica: considere a possibilidade de definir via pragmas até que extensão a parte da declaração da biblioteca deve estar visível no Gerenciador de Bibliotecas, quando ela for incluída em um projeto. ("Omitindo declarações de variáveis").

No caso de terem sido implementadas POUs em outras linguagens de programação, por exemplo, C, para colocá-las em uma biblioteca, então salve o projeto usando o tipo de dados **Biblioteca Externa * .lib.** Além do arquivo da biblioteca, estará disponível um arquivo com a extensão "*.h". Este arquivo é estruturado como um arquivo de cabeçalho C e contém as declarações de todas as POUs, tipos de dados e variáveis globais, que estão disponíveis com a biblioteca. Se uma biblioteca externa é usada em um projeto, então, no modo de simulação, essa implementação das POUs que foram escritas com o MasterTool IEC será executada; mas, no dispositivo, a implementação do C será processada.

Se desejar adicionar informações de licenciamento a uma biblioteca, pressione o botão **Editar informações de licenciamento**... e insira as configurações apropriadas na caixa de diálogo 'Editar informações de licenciamento'. Consultar a descrição correspondente no 'Arquivo' 'Salvar como... ' e no Gerenciamento de licenças no MasterTool IEC.

Menu 'Inserir' 'Biblioteca adicional'

Com esse comando pode-se anexar uma biblioteca adicional a seu projeto.

O comando abre a caixa de diálogo para abrir um arquivo. Se o diretório atualmente definido não contém a biblioteca desejada, pode-se selecionar outro diretório no campo **Diretório de bibliotecas**, onde todos os diretórios que são definidos no 'Projeto' 'Opções' 'Diretórios' 'Bibliotecas' (tipo de arquivo "* .**lib** ") serão oferecidos. Escolha a biblioteca desejada - seleção múltipla é possível - e confirme com OK. A caixa de diálogo será fechada e a biblioteca é inserida para o Gerenciador de Bibliotecas. Agora pode-se usar os objetos da biblioteca do projeto como objetos definidos pelo usuário.

Caminhos de biblioteca

Considere quais diretórios de bibliotecas estão atualmente definidos nas opções do projeto Categoria diretórios (veja capítulo 4). Ao inserir uma biblioteca de um diretório que não está definido ali, a biblioteca será inserida com o caminho respectivo.

Exemplo: inserção da biblioteca standard. lib do diretório "D:\mastertooliec\libraries\standard". Se esse diretório é definido nas opções do projeto, a entrada no Gerenciador de Bibliotecas será: "standard. lib <data e hora do arquivo>".

Se nas opções do projeto foi definido apenas um diretório "D:\mastertooliec\libraries", então a entrada no Gerenciador de Bibliotecas será: "standard\standard. lib <data e hora do arquivo>".

Se nenhum diretório foi definido nas opções de projeto, então, o caminho completo será inserido: "D:\mastertooliec\libraries\standard\standard. lib <data e hora do arquivo>".

Ao reabrir o projeto as bibliotecas serão pesquisadas de acordo com as entradas no Gerenciador de Bibliotecas. Portanto, por exemplo, se apenas o nome de arquivo da biblioteca é inserido, a biblioteca será pesquisada nos diretórios das bibliotecas definidos nas opções de projeto.

Se as bibliotecas não forem encontradas ao abrir um projeto, será solicitado se deseja-se alterar o diretório de biblioteca definido nas opções de projeto. Em caso negativo, uma caixa de diálogo será aberta fornecendo informações sobre as bibliotecas não encontradas e as respectivas entradas no Gerenciador de Bibliotecas serão exibidas na cor vermelha. Nesse caso pode-se selecionar uma entrada de cor vermelha e escolher o comando **Pesquisar** do menu de contexto. Assim, aparecerá a caixa de diálogo para abrir um arquivo onde pode-se navegar para a biblioteca que está faltando e recarregá-la imediatamente.

Licenciamento

Ao incluir uma biblioteca para a qual uma licença é necessária e nenhuma licença válida for encontrada, pode-se receber uma mensagem de que a biblioteca está disponível somente no modo de demonstração ou que a biblioteca não está licenciada para o dispositivo atualmente definido. Pode-se ignorar essa mensagem ou iniciar ações apropriadas relativas à licença. Uma licença inválida produzirá um erro durante a compilação ('Projeto' 'Compilar'). Nesse caso um clique duplo na mensagem de erro ou <F4> irá abrir a caixa de diálogo 'Informações sobre licenciamento' onde pode-se iniciar as ações apropriadas guiadas por um assistente.

Remover biblioteca

Com o comando 'Editar' 'Excluir' pode-se remover uma biblioteca de um projeto e do Gerenciador de Bibliotecas.

Menu ‘Extras’ ‘Propriedades’

Esse comando abre a caixa de diálogo 'Informações sobre biblioteca interna (externa)'. Para bibliotecas internas estão disponíveis lá todos os dados que foram inseridos nas Informações do Projeto (onde aplicável incluindo também as informações sobre licenciamento) quando a biblioteca foi criada no MasterTool IEC. Para bibliotecas externas, o nome da biblioteca e o caminho da mesma serão exibidos.

Configuração do CP

A configuração do CP encontra-se como um objeto no cartão de registro **Recursos** no Organizador de Objetos. Com o Editor de Configuração do CP, deve-se descrever o hardware para o qual o projeto aberto é estabelecido. Para a implementação do programa, o número e a posição das entradas e saídas é especialmente importante. Com esta descrição, o **MasterTool IEC** verifica se os endereços IEC usados no programa também existem no hardware.

A base para trabalhar no editor de configuração é/são os arquivos de configuração (*.cfg; veja abaixo as **anotações relativas à compatibilidade de versão**) e os arquivos de dispositivo (. por exemplo, *.gsd, *.eds). Estes são armazenados no diretório que é definido no arquivo do dispositivo (consultar configurações do dispositivo) e são lidos quando o projeto for aberto no MasterTool IEC. **Pode-se adicionar arquivos a estes diretórios a qualquer momento**.

O arquivo de configuração *.cfg descreve uma configuração básica, a qual é mapeada no editor de configuração e define em que medida o usuário pode personalizar esta configuração no editor.

Atenção: logo que o arquivo de configuração subjacente (*.cfg) foi modificado, deve-se refazer a configuração no MasterTool IEC!

Nota relativa à compatibilidade de versão: no MasterTool IEC V2.2 um novo formato foi implementado para a configuração do CP. A partir desta versão em diante, os arquivos de configuração básicos devem usar a extensão ***cfg** . Em contrapartida, o editor de configuração nas versões anteriores precisava de arquivos com uma extensão ***. con**. Porém: o sistema do dispositivo pode definir que o Configurador "antigo" deva ser usado, mesmo quando um antigo projeto for aberto na V2.2 ou posterior (considere as observações mais abaixo). Isso evita a necessidade de criar novos arquivos de configuração e os arquivos*.con podem ser usados. Se o dispositivo não suportar o configurador antigo, pode-se converter a antiga configuração do CP armazenada no projeto para o novo formato; se (!) um novo arquivo *.cfg adequado tiver sido fornecido. Ver mais detalhes em 'Extras' 'Converter', neste mesmo capítulo.

O editor de configuração do MasterTool IEC permite configurar módulos de E/S.

Se for suportado pelo sistema do dispositivo, existe a possibilidade de obter informações a partir do CP:

1. Uma verificação da estrutura atual do hardware que pode ser diretamente utilizada na configuração do CP
2. Mensagens de diagnóstico que serão exibidas como mensagens no MasterTool IEC
3. Informações de status que serão exibidas na caixa de diálogo Configuração do CP

Após a personalização final pelo usuário, uma imagem binária da configuração é enviada para o CP.

Exemplo de Configuração do CP:

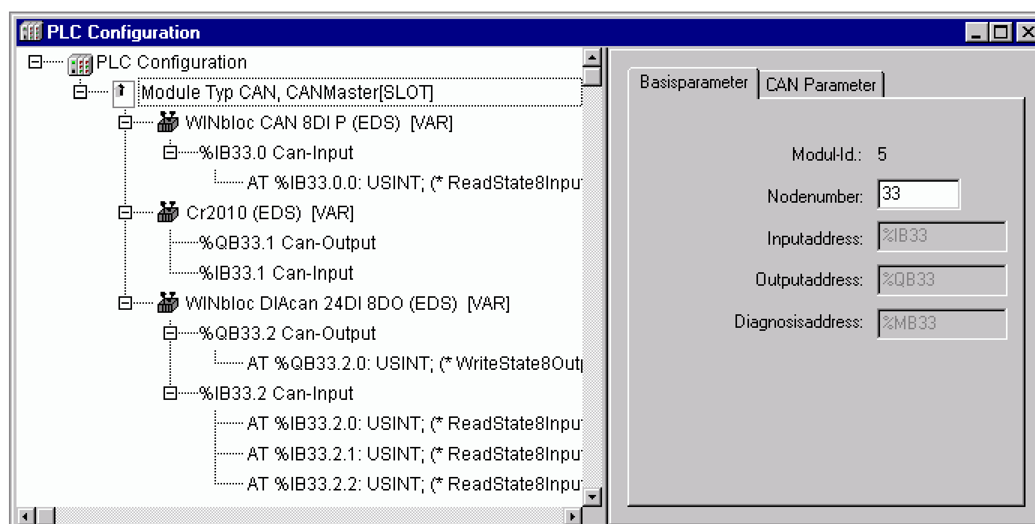


Figura 6-7. Exemplo de Configuração do CP

A configuração do CP é exibida no editor em estrutura de árvore e pode ser editada usando os comandos do menu e as caixas de diálogo. A configuração contém elementos de entrada e/ou saída e também elementos de gestão, os quais, por sua vez, também podem ter sub-elementos.

Para entradas e saídas, podem ser atribuídos nomes simbólicos. O endereço IEC onde esta entrada ou saída pode ser acessada, está localizado atrás do nome simbólico.

Visão geral da configuração

Trabalhando na configuração do CP

O editor de configuração é dividido em duas partes. Na janela à esquerda, a **árvore de configuração** é exibida. A estrutura e os componentes da árvore resultam principalmente (configuração padrão) das definições encontradas no arquivo de configuração, mas podem ser modificadas pela adaptação

adicional feita pelo usuário no Configurador do CP no MasterTool IEC. Na janela à direita, as **caixas de diálogo de configuração** atualmente disponíveis são mostradas em uma ou várias guias.

A parte direita da janela é, por padrão, visível, mas pode ser retirada via menu **'Extras' 'Propriedades'**.

Na parte superior da árvore de configuração está a entrada do **módulo "raiz**" com um nome, o qual foi definido no arquivo de configuração *.cfg.

A seguir encontram-se hierarquicamente recuados os outros elementos da configuração: módulos de diferentes tipos, canais ou canais de bits.

Seleção de elementos

Para selecionar elementos, clique com o mouse no elemento correspondente ou, usando as teclas de setas, mova o retângulo pontilhado até o elemento desejado.

Elementos que começam com um sinal de mais são elementos de organização e contêm sub-elementos. Para abrir um elemento, selecione o elemento e dê um clique duplo no sinal de adição ou pressione <Enter>. Pode-se fechar elementos abertos (sinal de menos na frente do elemento) da mesma maneira.

Inserir elementos, 'Inserir' 'Inserir elemento', 'Inserir' 'Acrescentar sub-elemento'

Dependendo das definições no (s) arquivo (s) de configuração e nos arquivos de dispositivo disponíveis, que tenham sido lidas quando o projeto estava aberto, uma composição básica de elementos é posicionada automaticamente na árvore de configuração. Se um desses elementos é selecionado, mais elementos podem ser adicionados, se isso for permitido pelas definições no arquivo de configuração e se os arquivos de dispositivo necessários estiverem disponíveis:

'Inserir' 'Inserir elemento': um elemento pode ser selecionado e inserido antes do elemento atualmente marcado na árvore de configuração.

'Inserir' 'Acrescentar sub-elemento': um elemento pode ser selecionado e inserido como sub-elemento do elemento que é marcado no momento na árvore de configuração. Ele será inserido na última posição.

Os comandos mais importantes são encontrados no menu de contexto (botão direito do mouse ou <Ctrl>+<F10>).

Nota: se suportado pelo sistema do dispositivo, uma verificação do hardware existente pode ser usada para a inserção dos módulos na configuração do CP no MasterTool IEC.

Substituir/trocar elementos, 'Extras' 'Substituir elemento'

Dependendo das definições no arquivo de configuração, o elemento atualmente selecionado pode ser substituído por outro. Da mesma maneira talvez seja possível trocar canais, que são configurados de tal forma que podem ser usados como elementos de entrada ou de saída. Use o comando 'Extras' 'Substituir elemento'.

Nomes simbólicos

Os nomes simbólicos para módulos e canais podem ser definidos no arquivo de configuração. Nesse caso eles serão mostrados no editor de configuração antes do 'AT' do endereço IEC do respectivo elemento. Na configuração do arquivo também é definido se o nome simbólico pode ser editado ou inserido no editor de configuração. Para inserir um nome simbólico, selecione o módulo desejado ou canal na árvore de configuração e abra um campo de texto com um clique do mouse em 'AT' antes do endereço IEC. Da mesma maneira, pode-se editar um nome simbólico existente após um duplo clique no nome. Considerar que alocar um nome simbólico corresponde a uma declaração de variável válida!

Exportação/importação de módulos

Se um módulo for definido como sendo "exportável" no arquivo de configuração (*.cfg), no menu de contexto estarão disponíveis os comandos 'Exportar módulo' e 'Importar módulo', quando o módulo for selecionado na árvore de configuração.

Com o comando 'Exportar módulo' a caixa de diálogo para selecionar um arquivo será aberta. Aqui pode-se definir um arquivo para o qual o módulo será exportado no formato XML junto com todos os sub-módulos e suas configurações. Esse arquivo pode ser importado em outra configuração do CP por meio do comando 'Importar módulo', se houver um módulo adequadamente definido selecionado na árvore de configuração.

Assim, de uma maneira fácil, a árvore de configuração de um módulo específico pode ser transferida para outra configuração de CP.

Ajustes gerais na configuração do CP

Selecione a entrada 'Configuração do CP' (módulo 'raiz') na parte superior da árvore de configuração. A caixa de diálogo 'Configurações' é mostrada na parte direita da janela. As opções a seguir podem ser ativadas:

Calcular endereços: cada módulo recém-inserido é alocado automaticamente com um endereço, o qual resulta do endereço do módulo inserido antes, mais o tamanho desse endereço. Se um módulo é removido da configuração, os endereços dos módulos a seguir são ajustados automaticamente. Quando o comando 'Extras' 'Computar endereços' for executado, todos os endereços começando no nó selecionado (módulo) serão recalculados.

Verificar sobreposição de endereços: na compilação o projeto será verificado se há sobreposição de endereços e uma mensagem correspondente será exibida.

Salvar arquivos de configuração no projeto: as informações contidas no(s) arquivo(s) de configuração *.cfg e nos arquivos de descrição do dispositivo que sustentam a configuração do CP atual, serão salvas no projeto.

Dessa forma (**se** não estiver definido pelo arquivo de configuração, que a **configuração padrão sempre seja recarregada**!), a configuração será mantida no projeto como foi definida pelo usuário, mesmo se arquivos de configuração não forem encontrados quando o projeto é reaberto. Lembre-se de que, neste caso, a configuração específica completa do projeto **será perdida**, se a opção descrita aqui não for ativada!

Ao salvar as informações de configuração com o projeto, elas também serão mantidas na alteração do dispositivo. Mas lembre-se que, nesse caso, o novo dispositivo pode colocar arquivos de configuração próprios, os quais serão considerados adicionalmente.

Caixa de diálogo para as Configurações Gerais da configuração do CP:

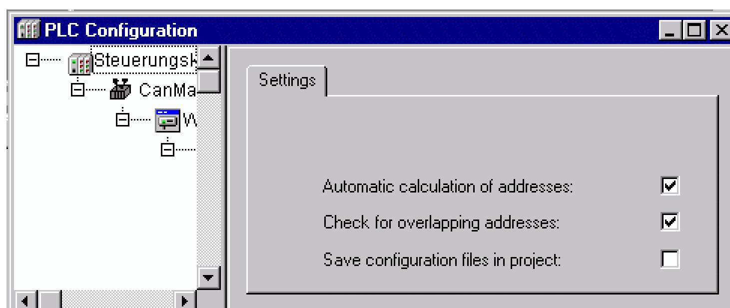


Figura 6-8. Caixa de diálogo para as Configurações Gerais da configuração do CP

O modo global de endereçamento (endereços de flat /endereços dependendo da ID) na configuração do CP é definido no arquivo de configuração.

Recálculo dos endereços de módulo, 'Extras' 'Calcular endereços'

Se a opção "Calcular endereços" está ativada no diálogo 'Configurações' do Editor de Configuração do CP, então, o comando 'Extras' 'Calcular endereços' será iniciado para recalcular os endereços dos módulos. Todos os módulos, começando com aquele que atualmente está selecionado na árvore de configuração, serão considerados.

Acrescentar arquivo de configuração

Use este comando no menu 'Extras' para adicionar um arquivo extra aos arquivos de configuração do projeto. Esses são arquivos encontrados no(s) caminho(s) do(s) diretório(s) especificado(s) nas opções de projeto, categoria 'Diretórios', campo de entrada: 'Arquivos de configuração'.

O diálogo **Selecionar arquivo de configuração** será aberto, onde pode-se definir um filtro para, arquivos de configuração (*.cfg) ou todos os arquivos (*. *). Depois de ter selecionado o arquivo desejado será verificado se o arquivo já está presente em um dos diretórios definidos para os arquivos de configuração. Neste caso uma mensagem apropriada será exibida e o arquivo não poderá ser adicionado. Se um arquivo tipo cfg for selecionado, uma caixa de diálogo será aberta, onde obtém-se informações sobre o que fazer.

Se o arquivo puder ser adicionado, o diálogo **Selecionar diretório de configuração**, onde todos os diretórios de configuração atualmente definidos para o projeto serão exibidos em uma lista de seleção. Escolha o diretório no qual o arquivo deve ser copiado. Após ter confirmado esta seleção pressionando o botão OK, a caixa de diálogo será fechada e o arquivo imediatamente estará disponível na configuração do CP.

Retornar a configuração padrão, 'Extras' 'Configuração padrão'

O comando 'Extras' 'Configuração padrão' pode ser usado para restaurar a configuração original do CP, que é definida pelo arquivo de configuração*.cfg e que está salva no projeto.

Atenção: no arquivo de configuração *.cfg pode ser definido que a configuração padrão deve ser restaurada **cada** vez que o projeto for reaberto. Nesse caso todas as adaptações da configuração feitas pelo usuário serão perdidas!

Diálogo de parâmetros específicos customizados

As possibilidades de parametrização do configurador podem ser expandidas com o uso de um aplicativo específico DLL, que é uma caixa de diálogo individual. Esta 'Hook'-DLL deve estar no diretório que contém o arquivo de configuração e, então, pode ser vinculada a um módulo ou canal através de uma entrada no arquivo de configuração. Assim sendo, para os módulos relativos à caixa de diálogo padrão 'Parâmetros de módulo' será substituída por uma caixa de diálogo definida na DLL.

Exemplo de uma caixa de diálogo de parâmetro de um aplicativo específico (Parâmetros personalizados):

Figura 6-9. Parâmetros personalizados

Configuração de um módulo de E/S

Parâmetros básicos de um módulo de E/S

Se um módulo de E/S é selecionado na árvore de configuração, a caixa de diálogo 'Parâmetros básicos' é exibida com as seguintes entradas...

ID do módulo: a identificação do módulo é um identificador exclusivo do módulo dentro de toda a configuração. Ele é definido pelo arquivo de configuração e não é editável no editor de configuração.

Número do nó: o número do nó é definido por uma entrada no arquivo de configuração ou – se não houver nenhuma entrada – pela posição do módulo na estrutura de configuração.

Endereço de entrada, **Endereço de saída, Endereço de diagnóstico**: endereços para entrada, saída e para o armazenamento de dados de diagnóstico.

Esses endereços referem-se ao módulo. Depende das configurações gerais, quais endereços já são predefinidos, qual modo de endereço é válido e se os endereços podem ainda ser editados aqui.

Comentário: aqui podem ser inseridas no módulo informações de texto adicionais.

Carregar descrição do módulo: se esta opção está desativada, o módulo não será considerado em um download do projeto (desativação de um módulo). Por padrão a opção é ativada e é definida no arquivo de configuração *.cfg se ele é visível e editável na caixa de diálogo de configuração.

Não adaptar endereço automaticamente: esta opção só esta disponível se definido pelo arquivo de configuração. Se ela está ativado, o módulo não será considerado em caso de um recálculo dos endereços (padrão: opção é desativada).

Caixa de diálogo de Parâmetros básicos para um módulo de E/S:

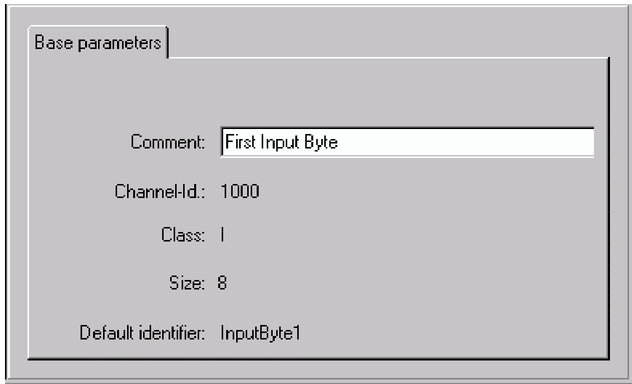


Figura 6-10. Parâmetros básicos para um módulo de E/S

O diagnóstico na configuração do CP:

O campo **Endereço de diagnóstico** serve para especificar um marcador de endereço para o qual as informações de diagnóstico (estado de barramento) serão gravadas automaticamente.

Para módulos de E/S normais vai depender da configuração especial do hardware como o diagnóstico será tratado.

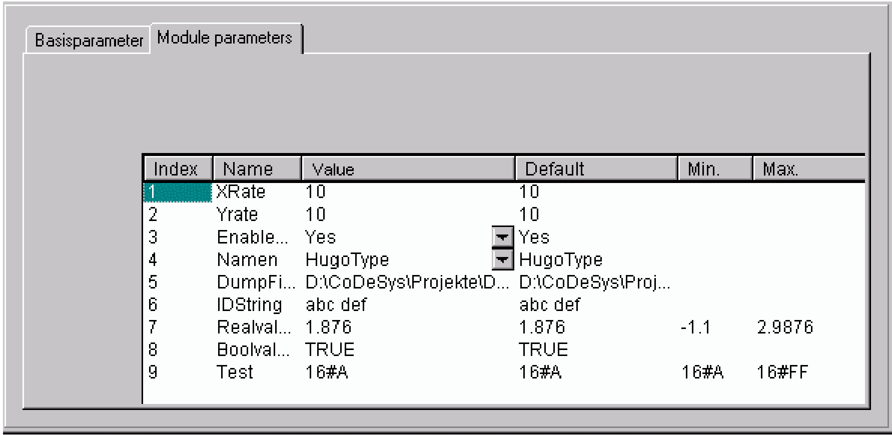
Para sistemas de barramento como CAN ou Profibus DP (consultar disponibilidade), o diagnóstico funciona como descrito a seguir: a partir do endereço de diagnóstico fornecido serão armazenadas várias informações relacionadas à **estrutura GetBusState** que faz parte de uma biblioteca correspondente fornecida pelo fabricante. Todos os módulos de barramento obtêm uma solicitação para preencher a estrutura de diagnóstico em uma seqüência cíclica a cada vez que os dados do processo de uma tarefa IEC tiverem sido escritos ou lidos a partir de/para os módulos.

Análise de diagnóstico via bloco funcional DiagGetState:

Assim que, pelo menos um módulo no sistema de barramento produzir um erro, as informações de diagnóstico específico poderão ser lidas usando o Bloco funcional DiagGetState, que também é parte da biblioteca mencionada acima. Esta função só está disponível para os barramentos mestres, a qual foi configurado dentro da configuração do CP no MasterTool IEC!

Parâmetros do Módulo/Parâmetros customizados de um módulo de E/S

Caixa de diálogo Parâmetros do módulo:



Index	Name	Value	Default	Min.	Max.
1	XRate	10	10		
2	Yrate	10	10		
3	Enable...	Yes	Yes		
4	Namen	HugoType	HugoType		
5	DumpFi...	D:\CoDeSys\Projekte\D...	D:\CoDeSys\Proj...		
6	IDString	abc def	abc def		
7	Realval...	1.876	1.876	-1.1	2.9876
8	Boolval...	TRUE	TRUE		
9	Test	16#A	16#A	16#A	16#FF

Figura 6-11. Caixa de diálogo Parâmetros do módulo

Nesta caixa de diálogo, os parâmetros que são fornecidos pelo arquivo de dispositivo são mostrados. Somente a coluna 'Valor' é editável.

Índice: o índice é um dígito consecutivo (i) que numera todos os parâmetros do módulo.

Nome: nome do parâmetro

Valor: Valor do parâmetro, editável

Inicialmente o padrão é exibido. Valores podem ser definidos diretamente ou por meio de nomes simbólicos. Se as entradas no arquivo de configuração não estiverem definidas como 'Somente leitura', elas podem ser editadas. Para fazer isto, clique no campo de edição e selecione uma das entradas em uma lista de rolagem. Se o valor for um nome de arquivo, pode-se abrir o diálogo 'Abrir arquivo' por um clique duplo e procurar outro arquivo ali.

Default: valor padrão dos parâmetros

Min.: valor mínimo do parâmetro (somente se não forem usados nomes simbólicos)

Max.: valor máximo do parâmetro (somente se não forem usados nomes simbólicos)

Uma tooltip pode fornecer informações adicionais sobre o parâmetro marcado no momento.

Em vez da caixa de diálogo de parâmetros do módulo, pode haver uma caixa de diálogo específica customizada. Isso se deve ao fato de que tal caixa de diálogo está vinculada por uma entrada (Hook-DLL) na definição do módulo no arquivo de configuração.

Configuração de um canal

Parâmetros Básicos de um canal

Canal-ID: identificador global exclusivo do canal

Classe: define se o canal é usado como entrada (I), saída (Q), ou como entrada e saída (I&Q), ou se ele é alternável (I|Q). Se o canal é alternável, isso pode ser feito pelos comando 'Extras' 'Substituir elemento'.

Tamanho: tamanho do canal [bytes]

Identificador default: nome simbólico do canal

O nome do canal é definido no arquivo de configuração. Somente se permitido pela definição do módulo pai, o nome do canal pode ser editado na árvore de configuração.

Comentário: informações adicionais sobre o canal

No campo de edição um comentário pode ser inserido ou modificado.

<u>Endereço</u>: este campo de edição só estará disponível se ele foi ativado por uma entrada no arquivo de configuração. Insira o endereço desejado para o canal.

Caixa de diálogo de parâmetros básicos de um canal:

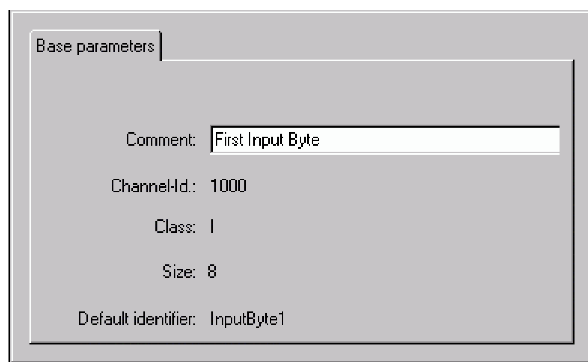


Figura 6-12. Caixa de diálogo de parâmetros básicos de um canal

Parâmetros de canal

Pertencente à caixa de diálogo de parâmetros do módulo, a caixa de diálogo de parâmetros do canal é usada para exibir e modificar os parâmetros de um canal: **índice, nome, valor,**

default, min., máx.. Esta caixa de diálogo também pode ser substituída por uma caixa de diálogo específica customizada ('Parâmetros customizados').

Canais de Bit

Canais de Bit são automaticamente inseridos quando um canal é definido com uma entrada CreateBitChannels=TRUE no arquivo de configuração.

A caixa de diálogo de parâmetros básicos de canais de bit contém apenas o campo de **comentário**.

Configuração de módulos MODBUS

Consultar a documentação específica referente à configuração de módulos MODBUS.

Configuração do CP no modo online

No modo online a configuração do CP exibe os estados das entradas e saídas do CP. Se uma entrada ou saída booleana têm o valor TRUE, a pequena caixa no início da linha de entrada na árvore de configuração ficará azul; valores não booleanos serão adicionados no final da linha de entrada (por exemplo, "=12").

As entradas booleanas podem ser alternadas por cliques de mouse. Em outras entradas um clique do mouse no início da linha abre uma caixa de diálogo, onde o valor pode ser modificado. O valor modificado será definido no CP assim que a caixa de diálogo é fechada com **OK**.

Também considere as possibilidades específicas do dispositivo para diagnósticos online.

Informações de hardware do CP

Se suportado pelo sistema do dispositivo e pelo arquivo de configuração atual (*.cfg), informações sobre a estrutura, status e resultados do diagnóstico da verificação do hardware conectado no momento, poderão ser obtidos a partir do CP e exibidos na configuração do CP no MasterTool IEC.

Verificação da configuração do módulo

Se o suportado pelo sistema do dispositivo e pelo arquivo de configuração atual (*.cfg), o comando Verificação da configuração do módulo estará disponível no menu de contexto para o módulo que atualmente está selecionado na árvore de configuração do CP.

Este comando só está disponível no modo offline. Se ele estiver ativado, a configuração de hardware atual do módulo específico no CP será verificada e será automaticamente oferecida para inserção na árvore de configuração da Configuração do CP no MasterTool IEC. Assim, a configuração do módulo existente facilmente pode ser mapeada no MasterTool IEC.

Carregar estado do módulo

Se o suportado pelo sistema do dispositivo e pelo arquivo de configuração atual (*.cfg), o comando **Carregar estado do módulo** estará disponível no menu de contexto para o módulo que atualmente está selecionado na árvore de configuração do CP.

Este comando só está disponível no modo online. Se ele estiver ativado, o status real do módulo será lido a partir do CP e será exibido em uma cor especial na árvore de configuração:

Preto: módulo existente e parametrizado corretamente.

Azul: módulo existente, mas parametrizado incorretamente.

Vermelho: módulo não encontrado

Uma atualização da exibição do status também será feita automaticamente a cada download.

Mostrar Mensagens de diagnóstico

Se o suportado pelo sistema do dispositivo e pelo arquivo de configuração atual (*.cfg), o comando **Mensagens de diagnóstico** estará disponível no menu de contexto para o módulo que atualmente

está selecionado na árvore de configuração do CP. Este comando só está disponível no modo online. Se ele estiver ativado, as mensagens de diagnóstico atuais para o módulo proveniente do CP serão exibidas em uma janela no MasterTool IEC.

Configuração de tarefas

Visão geral da configuração

Além de declarar o programa especial PLC_PRG, pode-se também controlar o processamento do projeto usando o Gerenciamento de Tarefas.

Uma **tarefa** é uma unidade de tempo no processamento de um programa IEC. Ela é definida por um nome, uma prioridade e por um tipo determinado de condição que irá disparar o início da mesma. Essa condição pode ser definida por um tempo (cíclico, execução livre) ou por um evento interno ou externo que irá disparar a tarefa; ex.: a borda de subida de uma variável global do projeto ou de um evento de interrupção do controlador.

Para cada tarefa, pode-se especificar uma série de programas que serão iniciados pela tarefa. Se a tarefa é executada no ciclo presente, esses programas serão processados no tamanho de um ciclo.

A combinação de prioridade e condição irá determinar em qual ordem cronológica as tarefas serão executadas.

Cada tarefa pode ser habilitada ou desabilitada explicitamente.

Para cada tarefa, pode-se configurar um cão de guarda (controle de tempo); as configurações possíveis dependem do sistema do dispositivo.

No modo online o processamento da tarefa pode ser monitorado em um diagrama.

Além disso, há a possibilidade de vincular **Eventos de Sistema** (por exemplo, Start, Stop, Reset) diretamente com a execução de uma POU do projeto.

A Configuração de tarefas é encontrada como um objeto na guia **Recursos** no Organizador de Objetos. O **Editor de tarefas** é aberto em uma janela bipartida.

Na parte esquerda da janela as tarefas são representadas em uma **árvore de configuração**. Na posição superior aparece a entrada 'Configuração de tarefas'. Abaixo estão as entradas 'Eventos de sistema' e as entradas para as tarefas específicas, representadas pelo nome da tarefa. Abaixo de cada entrada de tarefa atribuída, as chamadas do programa são inseridas. Cada linha é precedida por um ícone.

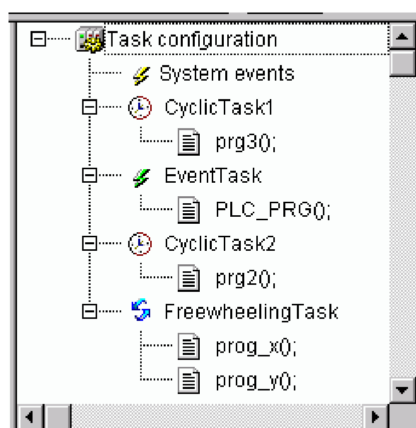


Figura 6-13. Exemplo de uma Configuração de tarefas

Na parte direita da janela, uma **caixa de diálogo** que pertence à entrada marcada atualmente na árvore de configuração será exibida. Aqui pode-se configurar as tarefas (Propriedades da tarefa), chamadas de programa (Chamada de programa) e definir a vinculação dos eventos de sistema

(Eventos de sistema). Depende do dispositivo quais opções estão disponíveis nas caixas de diálogo de configuração. Elas são definidas por um arquivo de descrição que é referido no arquivo do dispositivo. Se as descrições padrão são estendidas pelas Definições específicas do cliente, então, elas serão exibidas em uma guia adicional 'Parâmetros', na parte direita da janela.

Nota: não use a mesma função string (consultar standard. lib) em várias tarefas, porque isso pode causar falhas de programa por sobrescrição.

Trabalhando com a configuração de tarefas

Os comandos mais importantes podem ser obtidos a partir do menu de contexto (botão direito do mouse).

No título da configuração de tarefas estão as palavras "Configuração de tarefas". Se um sinal de mais está localizado antes das palavras, a lista seqüencial é fechada. Com um clique duplo na lista ou pressionando <Enter>, pode-se abrir a lista. Agora aparecerá um sinal de subtração. Com um clique duplo mais uma vez, pode-se fechar a lista novamente. Para cada tarefa, há uma lista de chamadas de programa anexada. Pode-se abrir e fechar essa lista da mesma maneira.

Com o comando 'Inserir' 'Inserir tarefa', pode-se inserir uma tarefa.

Com o comando 'Inserir' 'Anexar tarefa', pode-se inserir uma tarefa no final da árvore de configuração.

Com o 'Inserir' 'Inserir chamada de programa', uma chamada do programa será atribuída à tarefa que está selecionada atualmente na árvore de configuração.

Para cada entrada na árvore de configuração, uma caixa de diálogo de configuração aparecerá na parte direita da janela. Ali, as opções podem ser ativadas/desativadas e entradas para o campo de edição podem ser feitas. Dependendo de qual entrada está selecionada na árvore de configuração, haverá a caixa de diálogo para definir os 'Atributos da tarefa' (consultar 'Inserir tarefa'), a caixa de diálogo para definir uma 'Chamada de programa' (consultar 'Inserir chamada de programa') ou a tabela de 'Eventos do sistema’. As opções de configuração dependem do sistema do dispositivo. As configurações feitas nas caixas de diálogo serão levadas para a árvore de configuração, assim que o foco for definido para a árvore novamente.

Um nome de tarefa ou nome de programa também pode ser editado na árvore de configuração. Para isso clique com o mouse no nome ou selecione a entrada e pressione o botão <espaço> para abrir um quadro de edição.

Pode-se usar as teclas de setas para selecionar a entrada anterior ou a próxima na árvore de configuração.

Clicando no nome de tarefa ou do programa, ou pressionando a <barra de espaço>, pode-se definir uma caixa de controle de edição em torno do nome. Então, pode-se alterar a designação diretamente no Editor de tarefas.

'Inserir' 'Inserir tarefa' ou 'Inserir' 'Anexar tarefa'

Com esse comando pode-se inserir uma nova tarefa para a Configuração de Tarefas. As entradas de cada uma consistem em um símbolo e em um nome da tarefa.

Se uma tarefa ou a entrada 'Eventos do sistema' é selecionada, o comando **'Inserir tarefa'** estará à sua disposição. A nova tarefa será inserida após aquela selecionada. Se a entrada 'Configuração de tarefas' for selecionada, então, **'Anexar tarefa'** está disponível e a nova tarefa será anexada ao final da lista existente. O número máximo de tarefas é definido pelo sistema do dispositivo. Considerar que um determinado número de tarefas já pode estar reservado para módulos de configuração do CP (definido no arquivo-cfg).

Ao inserir uma tarefa, a caixa de diálogo para definir os atributos da tarefa será aberta.

Insira os atributos desejados...

Nome: um nome para a tarefa; com esse nome a tarefa é representada na árvore de configuração; o nome pode ser editado após um clique do mouse na entrada ou depois de pressionar a tecla <espaço> quando a entrada está selecionada.

Prioridade (0-31): (um número entre 0 e 31; 0 é a prioridade mais alta, 31 é a menor),

Caixa de Diálogo para Configuração de Atributos de Tarefas:

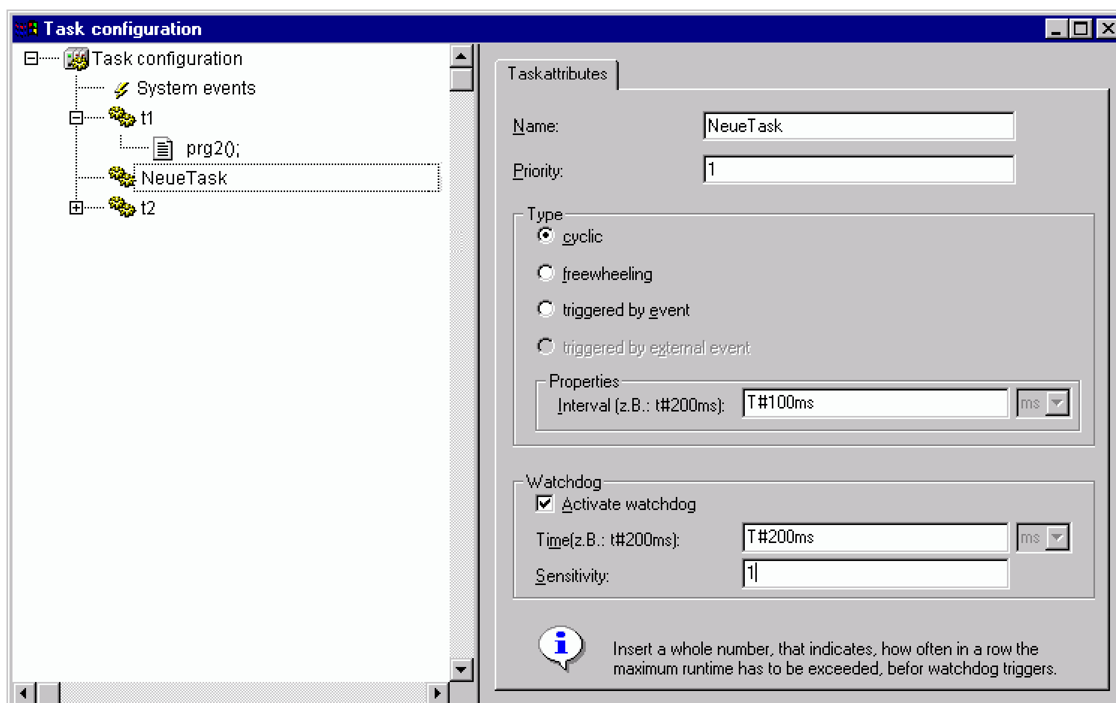


Figura 6-14. Caixa de Diálogo para Configuração de Atributos de Tarefas

Tipo

Cíclica: a tarefa será processada ciclicamente de acordo com a definição de tempo dada no campo 'Intervalo' (veja abaixo).

Execução livre: a tarefa será processada assim que o programa é iniciado e no final de uma execução será reiniciado automaticamente, em um laço contínuo. Não há tempo de ciclo definido.

Disparada por evento: a tarefa será iniciada na borda de subida da variável que é definida no campo Evento.

Disparada por evento externo: a tarefa será iniciada assim que o evento de sistema, que é definido no campo Evento, ocorre. Depende do dispositivo, quais eventos serão suportados e oferecidos na lista de seleção (não devem ser confundidos com eventos do sistema)

Propriedades para tarefas disparadas por evento

Intervalo (para o tipo 'cíclica' e 'disparada por evento externo' se o evento requer uma definição de tempo): o período de tempo, após o qual a tarefa deve ser reiniciada. Ao digitar um número, pode-se escolher a unidade desejada na caixa de seleção atrás do campo de edição: milissegundos [ms] ou microssegundos [µs]. As entradas em formato [ms] serão mostradas no formato TIME (por exemplo, "t #200ms") assim que a janela for redesenhada; mas também pode-se inserir diretamente o valor no formato específico de tempo. As entradas em [ms] sempre serão exibidas como um número puro (por exemplo, "300").

Evento (para o tipo 'disparada por evento ' ou ' disparada por evento externo'): uma variável global irá disparar o início da tarefa, assim que uma borda de subida for detectada. Use o botão “...” ou o assistente de entrada <F2> para obter uma lista de todas as variáveis globais disponíveis.

Possivelmente, o sistema do dispositivo define eventos Singleton. Esses são eventos que permitem somente iniciar uma tarefa simples. Se um evento inicia diversas tarefas, será verificado durante a compilação do projeto. A verificação considera o endereço de dados de variável do evento, não no

nome. Por exemplo: se o sistema do dispositivo define %MX1. 1 e %IB4 como eventos Singleton, usando as seguintes variáveis como variáveis de evento ele produzirá dois erros (a e b, bem como c e d, cada um com o mesmo endereço).

```
VAR_GLOBAL
A AT %MX1.1: BOOL;
B AT %MX1.1: BOOL;
C AT %MB4: BOOL;
D AT %MD1: BOOL;
END_VAR
```

Se não há nenhuma entrada em ambos os campos 'Intervalo' e 'Evento', então, o intervalo da tarefa dependerá de qual controlador é usado (consultar a documentação do controlador); por exemplo, nesse caso para MasterTool IEC SP NT V2.2 e superior serão usados intervalos de 10 ms).

Cão de guarda

Para cada tarefa pode ser configurado um controle de tempo (cão de guarda). Se o sistema do dispositivo suportar uma configuração de cão de guarda estendida, possivelmente existem limites superiores, inferiores e um limite padrão predefinidos para o tempo de cão de guarda, bem como uma definição de tempo em porcentagem.

Ativar cão de guarda: quando esta opção é ativada, considerando a sensibilidade atualmente definida (consultar abaixo), a tarefa será encerrada no status de erro, assim que o processamento demorar mais do que definido no campo 'Tempo' (veja abaixo).

Atenção: sistema do dispositivo MasterTool IEC SP 32 bit Full desativa a função de cão de guarda quando o controle de fluxo é ativo ou quando a execução está parada no momento em um breakpoint.

Tempo (ex.: t#200ms): tempo de monitoração; após a expiração desse tempo, em relação à sensibilidade atualmente definida (consultar abaixo), o cão de guarda será ativado, a menos que a tarefa não tenha sido finalizada. Dependendo do sistema do dispositivo o tempo deve ser inserido como porcentagem do intervalo da tarefa. Nesse caso, a caixa de seleção de unidade é cinza e mostra "%".

Sensitividade: aqui pode-se inserir, em números inteiros, para qual saturação do tempo de cão de guarda, um erro deve ser gerado. A entrada padrão é "1", isto é, na primeira saturação do tempo de cão de guarda ocorrerá um erro. Atenção: se “0" é inserido, o cão de guarda será desativado!

Atributos específicos do fabricante

Além desses atributos padrão para a tarefa selecionada no momento, atributos específicos do fabricante podem ser exibidos em uma segunda guia "**Parâmetros**". Este será o caso se definido no arquivo de descrição do dispositivo específico para a configuração da tarefa.

'Inserir' 'Inserir chamada de programa' ou 'Inserir' 'Anexar chamada de programa'

Com esses comandos aparecerá a caixa de diálogo para inserir uma chamada de programa para uma tarefa na configuração de tarefas. Cada entrada na árvore de configuração de tarefa consiste em um símbolo apropriado e o nome do programa.

Com **'Inserir chamada de programa'**, a nova chamada do programa é inserida antes da chamada do programa selecionado e com **'Anexar chamada de programa'** a chamada do programa é acrescentada ao final da lista existente (ou chamadas de programa).

Caixa de diálogo para entrada de chamada de programa:

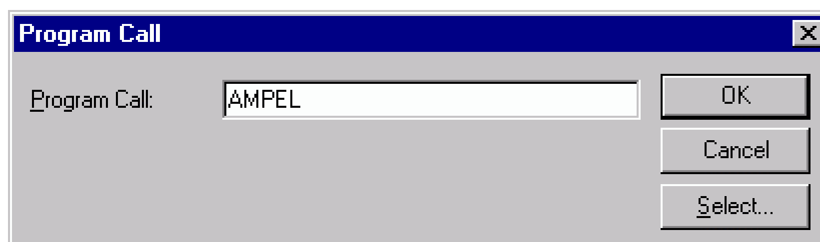


Figura 6-15. Caixa de diálogo para entrada de chamada de programa

No campo 'chamada de programa' especifique um nome de programa válido do seu projeto ou abra o assistente de entrada com o botão **Selecionar**, para selecionar um nome de programa válido. O nome do programa mais tarde também pode ser modificado na árvore de configuração. Para isso selecione a entrada e pressione a tecla <espaço> ou execute apenas um clique do mouse para abrir um campo de edição. Se o programa selecionado requer variáveis de Entrada, então, insira essas em sua forma usual e o tipo declarado (por exemplo, prg (invar:=17)).

O processamento das chamadas de programa mais tarde no modo online será feito de acordo com a sua ordem (de cima para baixo) no Editor de tarefas.

Considerar: não use a mesma função string em várias tarefas (consultar Elementos de biblioteca padrão), porque nesse caso valores podem ser sobrescritos durante o processamento das tarefas.

Eventos de sistema

Em vez de uma "tarefa", um "evento de sistema" pode ser usado para chamar uma POU do seu projeto. Os eventos de sistema disponíveis são específicos do dispositivo (definição no arquivo do dispositivo). A lista de eventos padronizados do dispositivo pode ser estendida por eventos específicos do cliente. Eventos possíveis são, por exemplo, Stop, Start, Alteração online.

A atribuição de eventos do sistema para POUs também é feita no Editor de configuração de tarefas. Use a caixa de diálogo 'Eventos', a qual será aberta assim que a entrada "Eventos de sistema" for selecionada na árvore de configuração de tarefas.

Tabela para atribuir POUs a Eventos de sistema:

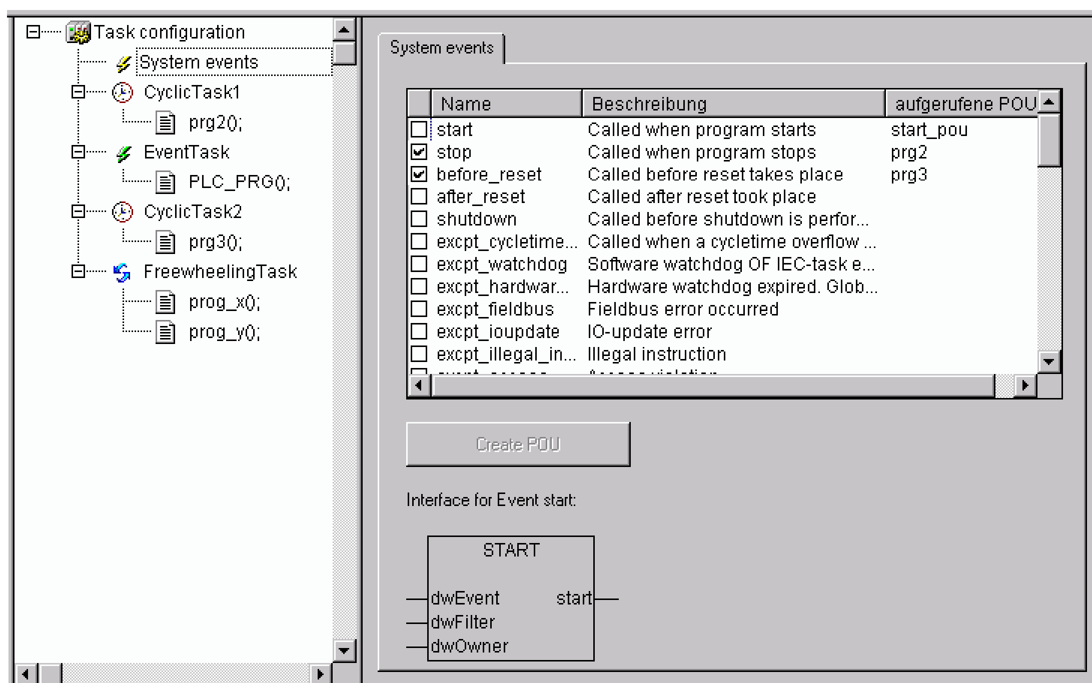


Figura 6-16. Tabela para atribuir POUs a Eventos de sistema

Cada evento é representado em uma linha: **nome** e **descrição** são exibidos conforme definido no arquivo do dispositivo, na coluna **chamada POU** pode-se inserir o nome do projeto POU que deve ser chamado e processado assim que o evento ocorrer.

Para isso use o assistente de entrada (<F2>) ou digite manualmente o nome de uma POU já existente (por exemplo, "PLC_PRG" ou "PRG.ACT1") ou insira um nome para uma POU não ainda existente.

Atenção para sistemas de dispositivo RISC e 68 K Motorola: o nome de uma função atribuída a um evento de sistema (função de retorno de chamada) **deve** começar com "callback"!

Para obter uma nova POU (função) criada no projeto, pressione o botão **Criar POU**. A POU será inserida no Organizador de objetos. Os parâmetros de entrada e saída que são requeridos para o evento serão automaticamente definidos na parte de declaração da POU. Abaixo da tabela de atribuição, o evento selecionado no momento é exibido em uma imagem, mostrando os parâmetros requeridos.

Se for necessário que a POU seja chamada pelo evento, ative a entrada na tabela de atribuição. A **ativação/desativação** é feita por um clique no mouse, na caixa de controle.

Configuração de tarefas no modo online

No modo online, o status e o número de ciclos de cada tarefa serão exibidos na árvore de configuração. O fluxo de tempo é monitorado em um diagrama. Pré-requisito: as bibliotecas SysTaskInfo.lib e SysLibTime.lib devem estar incluídas no projeto para fornecer funções para a avaliação interna do tempo das tarefas. As bibliotecas serão incluídas automaticamente assim que um dispositivo que suporta a monitoração de tarefas seja definido.

Exibição do status da tarefa na árvore de configuração

No modo online, o status atual de uma tarefa será exibido entre parênteses no final da linha de entrada da árvore de configuração, assim como também o número de tarefas já passado por meio de ciclos do processo. Esse intervalo de atualização é o mesmo utilizado para a monitoração dos valores do CP. Status possíveis:

- **Ociosa -** Não foi iniciada desde a última atualização; especialmente usada para tarefas de eventos
- **Em execução -** Foi iniciada pelo menos uma vez desde a última atualização
- **Parada** - Interrompida

- **Parada em BP -** Interrompido, porque o breakpoint na tarefa foi atingido
- **Parada no erro -** Erro, por exemplo, uma divisão por zero, falha de página, etc.
- **Parada por cão de guarda -** Tempo do ciclo foi excedido

A entrada da tarefa será exibida em cor vermelha no caso do status 'Parada no erro' ou 'Parada por cão de guarda'.

Exibição do fluxo de tempo das tarefas:

Se a entrada 'Configuração de tarefas' está selecionada na árvore de configuração, a utilização das tarefas será exibida em gráficos de barras na parte direita da janela:

Exibição da execução de tarefas no modo online:

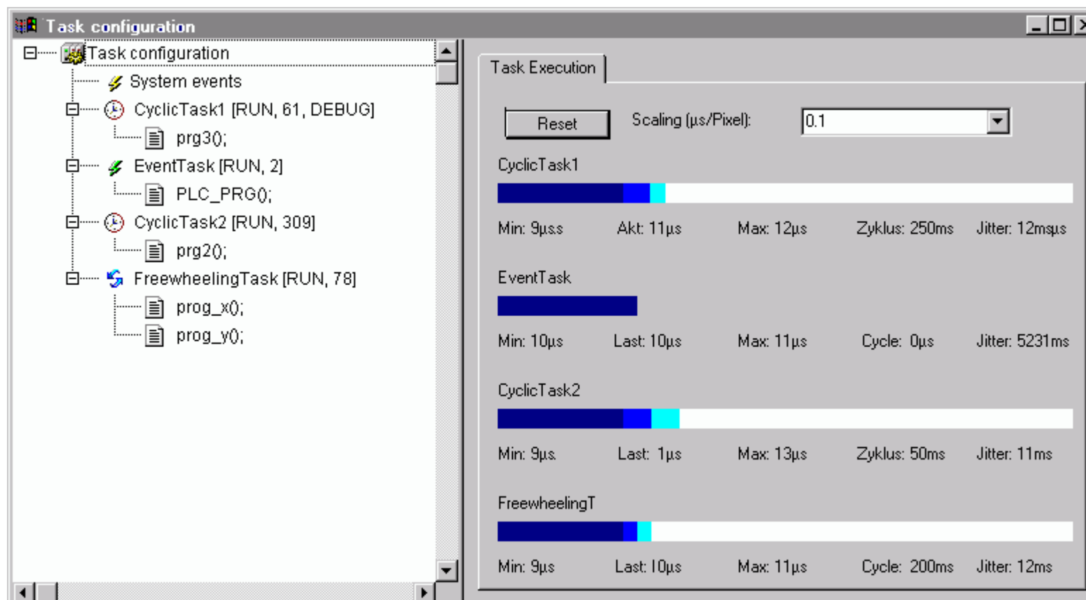


Figura 6-17. Exibição da execução de tarefas no modo online

Para cada tarefa é exibido um gráfico de barras. O comprimento da barra representa o tamanho de um período do ciclo. Abaixo da barra, bem como por marcas apropriadas na barra, os seguintes valores de medida são ilustrados:

- **Min -** tempo de execução mínimo medido em µs
- **Akt -** último tempo de execução medido em µs
- **Max -** tempo de execução máximo medido em µs
- **Ciclo -** comprimento total de um ciclo em µs
- **Jitter -** máximo jitter (tempo entre o início da tarefa e a indicação de que o sistema operacional está sendo executado) medido em µs.

O botão Reset pode ser usado para definir novamente os valores de Min., Max. e atenuação para 0.

A escala do gráfico (microssegundos por pixel) pode ser ajustada com a ajuda de uma lista de seleção em **Escala [µs/pixel]**.

Qual tarefa está sendo processada?

Para a execução, as seguintes regras se aplicam:

- A tarefa executada é aquela cuja condição foi atendida; isto é, se seu tempo especificado tiver expirado ou após sua variável de condição (evento) apresentar uma borda de subida.
- Se várias tarefas têm um requisito válido, então, a tarefa com a prioridade mais alta será executada.
- Se várias tarefas têm condições válidas e prioridades equivalentes, então a tarefa que teve o maior tempo de espera será executada primeiro.
- O processamento de chamadas de programa será executado de acordo com a sua ordem (de cima para baixo) no editor de tarefas.

- Dependendo do sistema do dispositivo, PLC_PRG pode ser processado em qualquer caso como uma tarefa de execução livre, sem que seja inserido na árvore de configuração de tarefas.

Funções online adicionais no menu de contexto e no menu 'Extras':

'Extras' 'Ativar tarefa de depuração'

Com esse comando uma tarefa de depuração pode ser submetida a SET no modo online na configuração de tarefas. O texto [DEBUG] será exibido após a tarefa definida.

Os recursos de depuração aplicam-se, então, somente a esta tarefa. Em outras palavras, o programa só pára em um breakpoint se o programa foi executado por meio da configuração de tarefas.

'Extras' 'Pilha de chamadas'

Este comando está disponível no menu Extras na configuração de tarefas. Se o programa for interrompido em um breakpoint, durante a depuração, ele pode ser usado para mostrar a pilha de chamadas da POU correspondente. Para essa finalidade a tarefa de depuração deve ser selecionada na árvore de configuração de tarefas. A janela 'Pilha de chamadas da tarefa <nome da tarefa>' será aberta. Lá aparece o nome da POU e a posição do breakpoint (por exemplo, "prog_x (2)" para a linha 2 da POU prog_x). Logo abaixo, a pilha de chamadas completa é mostrada na ordem inversa. Ao pressionar o botão 'Ir para', o foco saltará para essa posição na POU que atualmente está marcada na pilha de chamadas.

Gerenciador de Monitoração e Receitas

Visão geral da configuração

Com a ajuda do Gerenciador de Monitoração e Receitas, pode-se exibir os valores das variáveis selecionadas. O Gerenciador de Monitoração e Receitas também torna possível pré-definir as variáveis com valores definitivos e transferi-los como um grupo para o CP ('Escrever receita'). Da mesma forma, valores atuais do CP podem ser lidos e armazenados no Gerenciador de Monitoração e Receitas ('Ler receita’). Essas funções são úteis, por exemplo, para configurar e inserir de parâmetros de controle.

Todas as listas de inspeção criadas ('Inserir' 'Nova lista de monitoração') são indicadas na coluna esquerda do Gerenciador de Monitoração e Receitas. Essas listas podem ser selecionadas com um clique do mouse ou uma tecla de seta. Na área à direita do Gerenciador de Monitoração e Receitas, as variáveis aplicáveis a qualquer momento são indicadas.

Para trabalhar com o Gerenciador de Monitoração e Receitas, abra o objeto para o **Gerenciador de Monitoração e Receitas** no cartão de registro Recursos no Organizador de Objetos.

Gerenciador de Monitoração e Receitas no modo offline

No modo offline, pode-se criar várias listas de inspeção no Gerenciador de Monitoração e Receitas usando 'Inserir' 'Nova lista de monitoração'.

Para inserir as variáveis a serem monitoradas, pode-se chamar uma lista de todas as variáveis com o Assistente de Entrada, ou pode-se inserir as variáveis com o teclado, de acordo com a notação a seguir:

<Nome da POU>.<Nome da variável>

Com variáveis globais, o nome da POU não é indicado. A indicação inicia com um ponto. O nome de variável pode conter vários níveis. Endereços podem ser inseridos diretamente.

Exemplo de uma variável de vários níveis:

PLC_PRG.INSTANCE1.INSTANCE2.STRUCTURE.COMPONENTNAME

Exemplo de uma variável global:

.GLOBAL1.COMPONENT1

Gerenciador de Monitoração e Receitas no modo offline:

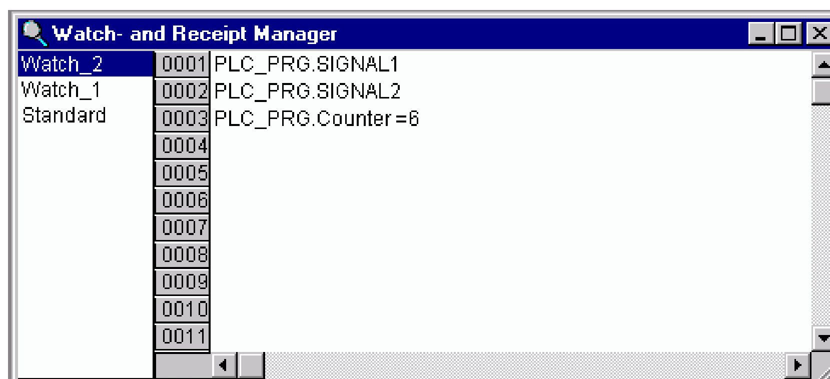


Figura 6-18. Gerenciador de Monitoração e Receitas no modo offline

As variáveis na lista de monitoração podem estar pré-definidas com valores constantes. Isto significa que, no modo online, pode-se usar o comando 'Extras' 'Escrever receita' para escrever esses valores nas variáveis. Para fazer isto, deve-se usar := para atribuir o valor constante na variável.

Exemplo:

PLC_PRG.TIMER:=50

Aqui, a variável PLC_PRG.TIMER é pré-definida com 50; no exemplo mostrado na imagem acima, a variável PLC_PRG.COUNTER é pré-definida com o valor 6.

Considere para variáveis do tipo **matriz** ou **estrutura**: deve-se inserir os elementos específicos explicitamente para poder defini-los previamente. Exemplo: ao definir uma estrutura com componentes a, b, c e declarar uma variável de estrutura struvar no PLC_PRG, para pré-alocar a, b e c com valores, digite-os na lista de monitoração da seguinte maneira:

PLC_PRG.struvar.a:=<valor>

PLC_PRG.struvar.b:=<valor>

PLC_PRG.struvar.c:=<valor>

A pré-definição para os elementos de uma matriz devem ser feitos correspondentemente. Exemplo de uma variável de matriz desse tipo ARRAY [0... 6]:

PLC_PRG.arr_var[0]:=<valor>

PLC_PRG.arr_var[1]:=< valor >

...Se um bloco funcional fb contém as variáveis x, y e uma variável de instância, fb_inst do tipo fb é declarada no PLC_PRG, então x e y podem estar pré-definidos da seguinte maneira:

PLC_PRG.fb_inst.x:=< valor >

PLC_PRG.fb_inst.y:=< valor >

'Inserir' 'Nova lista de monitoração'

Com este comando, no modo offline, uma nova lista de monitoração pode ser inserida no Gerenciador de Monitoração e Receitas. Digite o nome desejado para a lista de monitoração na caixa de diálogo que aparecer.

'Extras' 'Renomear lista de monitoração'

Com esse comando pode-se alterar o nome de uma lista de monitoração no Gerenciador de Monitoração e Receitas.

Na caixa de diálogo que aparecer, digite o novo nome da lista de monitoração.

'Extras' 'Salvar lista de monitoração'

Com esse comando pode-se salvar uma lista de monitoração. A caixa de diálogo para salvar um arquivo é aberta. O nome do arquivo é pré-definido com o nome da lista de monitoração e é fornecido com a extensão "*.wtc".

A lista de monitoração que foi salva pode ser carregada novamente com 'Extras' 'Carregar lista de monitoração'.

Gerenciador de Monitoração e Receitas no modo online

No modo online, os valores das variáveis inseridas são indicados.

Valores estruturados (matrizes, estruturas ou instâncias de blocos funcionais) são marcados por um sinal de mais na frente do identificador. Clicando no sinal de adição com o mouse ou pressionando <Enter>, a variável é aberta ou fechada.Se uma variável de bloco funcional é marcada na lista de monitoração, o menu de contexto associado é expandido para incluir os dois itens do menu 'Zoom' e 'Abrir instância'.

Para inserir novas variáveis, pode-se desativar a exibição usando o comando 'Extras' 'Ativar monitoração'. Depois que as variáveis foram inseridas, pode-se usar o mesmo comando novamente para ativar a exibição de valores.

Gerenciador de Monitoração e Receitas no modo online:

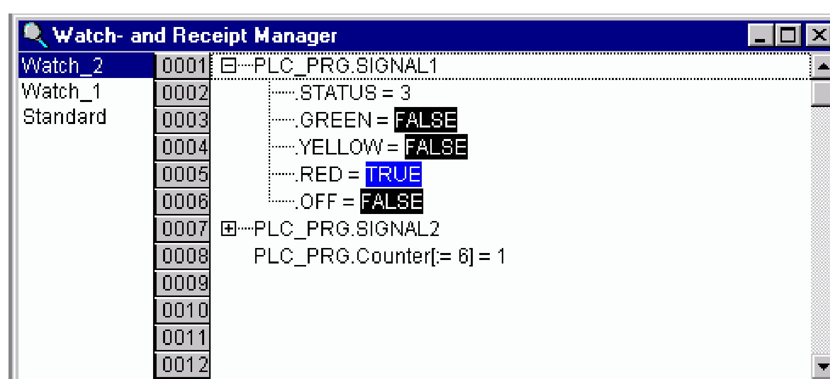


Figura 6-19. Gerenciador de Monitoração e Receitas no modo online

No modo offline, pode-se pré-definir variáveis com valores constantes (inserindo := <valor> após a variável). No modo online, esses valores agora podem ser escritos nas variáveis, usando o comando 'Extras' 'Escrever receita'.

Em referência a variáveis de matriz ou estrutura, por favor, veja a descrição neste mesmo capítulo.

Com o comando 'Extras' 'Ler receita', pode-se substituir a pré-definição da variável com o valor presente da variável.

Nota: somente esses valores da lista de monitoração que foram selecionados no Gerenciador de Monitoração e Receitas são carregados!

'Extras' 'Monitoração ativa'

Com esse comando no Gerenciador de Monitoração e Receitas no modo online, a exibição é ativada ou desativada. Se a exibição estiver ativa, um sinal de verificação (✓) aparecerá na frente do item do menu.

Para inserir novas variáveis ou para pré-definir um valor (consultar Modo offline), a exibição deve ser desativada por meio do comando. Depois que as variáveis foram inseridas, pode-se usar o mesmo comando novamente para ativar a exibição de valores.

'Extras' 'Escrever receita'

Com esse comando no modo online do Gerenciador de Monitoração e Receitas pode-se escrever a pré-definição dos valores (consultar Modo offline) para as variáveis.

Nota: somente os valores da lista de monitoração que foram selecionados no Gerenciador de Monitoração e Receitas são carregados!

'Extras' 'Ler receita'

Com esse comando no modo online do Gerenciador de Monitoração e Receitas pode-se substituir a pré-definição das variáveis (consultar Modo offline) pelo seu valor atual.

Exemplo:

PLC_PRG. COUNTER [:= <P. V.>] = <P. V.>

Nota: somente os valores da lista de monitoração que foram selecionados no Gerenciador de Monitoração e Receitas são carregados!

Forçar valores

No Gerenciador de Monitoração e Receitas, também é possível **'Forçar valor(es)'** e **'Escrever valor(es)'**. Ao clicar no respectivo valor da variável, então, uma caixa de diálogo é aberta, na qual pode-se inserir o novo valor da variável. Variáveis alteradas aparecem em vermelho no Gerenciador de Monitoração e Receitas.

Área de trabalho

Este objeto na guia 'Recursos' fornece uma imagem das opções atualmente definidas no projeto. Abrindo-a, obtém-se a caixa de diálogo 'Opções' com as categorias conhecidas.

Gerenciador de parâmetros

O gerenciador de parâmetros é um componente específico do dispositivo do Sistema de Programação MasterTool IEC e deve ser ativado nas configurações do dispositivo.

O gerenciador de parâmetros pode ser usado para tornar as variáveis de um programa IEC do MasterTool IEC, parâmetros constantes ou parâmetros do sistema específico acessíveis a todos os sistemas compatíveis com o MasterTool IEC em uma rede com a finalidade de troca de dados, geralmente via Field bus. Para essa finalidade, no editor, pode-se criar listas de parâmetros e lê-las ou enviá-las para o controlador.

Considerar: listas de parâmetros também podem ser criadas e preenchidas com entradas via pragmas que estão incluídas em declarações de variáveis (consultar capítulo 5: Instruções de Pragma)

O que são parâmetros?

Neste contexto parâmetros são:

- Variáveis do processo no projeto MasterTool IEC
- Parâmetros independentes do processo
- Parâmetros do sistema específico, predefinidos pelo sistema do dispositivo
- Instâncias blocos funcionais ou estruturas, variáveis, matrizes

Cada parâmetro é identificado por um determinado conjunto de **atributos**, como por exemplo, 'valor padrão', 'direitos de acesso' e, principalmente, por uma **chave de acesso** exclusivo ('Índice', 'Sub-índice', 'Nome'), que podem ser usadas para ler ou escrever dados de/para a lista de parâmetros. Esse intercâmbio de dados pode ser feito por meio de serviços de comunicação e não é necessário conhecer os endereços de variáveis ou fornecer quaisquer funções extras. Portanto, o uso da funcionalidade do Gerenciador de Parâmetros é uma alternativa para usar Variáveis de Rede.

O que são listas de parâmetros?

Listas de parâmetros são usadas para organizar os parâmetros e podem ser salvas com o projeto e carregadas para o sistema do dispositivo local que é controlado pelo programa IEC correspondente. Para cada tipo de parâmetros há um tipo correspondente de lista de parâmetros.

Cada entrada de parâmetro é representada por uma linha na lista de parâmetros. Cada **coluna** da lista está representando um dos atributos de parâmetro. Em acréscimo a um determinado conjunto de atributos padrão, também atributos específicos do fabricante podem ser usados para a descrição de um parâmetro no Gerenciador de Parâmetros.

Depende das definições em um **arquivo de descrição específico do dispositivo** quais atributos (colunas) serão visíveis e editáveis no Gerenciador de parâmetros e de que maneira elas serão organizadas em uma lista de parâmetros. Se o arquivo Descrição estiver faltando, o conjunto completo de atributos padrão será exibido, mostrando o valor padrão.

Editor do Gerenciador de parâmetros do MasterTool IEC:



Figura 6-20. Editor do Gerenciador de parâmetros do MasterTool IEC

Além das listas para variáveis de projeto e constantes de projeto, o Gerenciador de Parâmetros também pode gerenciar listas de parâmetros do sistema. Esses são os parâmetros que são fornecidos pelo sistema do dispositivo. Mais adiante pode-se criar listas para instâncias bloco funcional ou variáveis de estrutura que se baseiam em **modelos** definidos pelo usuário também criados no Gerenciador de parâmetros.

Devido ao fato de que os dados são armazenados independentemente do programa IEC, uma lista de parâmetros, por exemplo, pode ser usada para salvar 'receitas', que são preservadas, mesmo se o programa é substituído por outra versão. Mais adiante, um CP em execução pode ser "alimentado" com receitas diferentes sem a necessidade de um novo download do programa.

Nota: depende do sistema do dispositivo, se o Gerenciador de Parâmetros vai ser considerado na criação de um projeto de **inicialização**.

Ativação do gerenciador

O gerenciador de parâmetros deve ser ativado na guia Recursos no MasterTool IEC nas configurações do dispositivo, categoria funcionalidade de rede. Aqui também, os intervalos de índice e sub-índice para as entradas nas listas de parâmetros do tipo 'Parâmetro' (constantes), 'Variáveis' (variáveis do projeto) e - se suportado pelo sistema do dispositivo - Mapeamentos (para PDOs de dispositivos CAN) devem ser definidos. Depende do sistema do dispositivo se essas configurações são visíveis e editáveis pelo usuário.

Ativando Gerenciador de Parâmetros na caixa de diálogo do sistema do dispositivo:

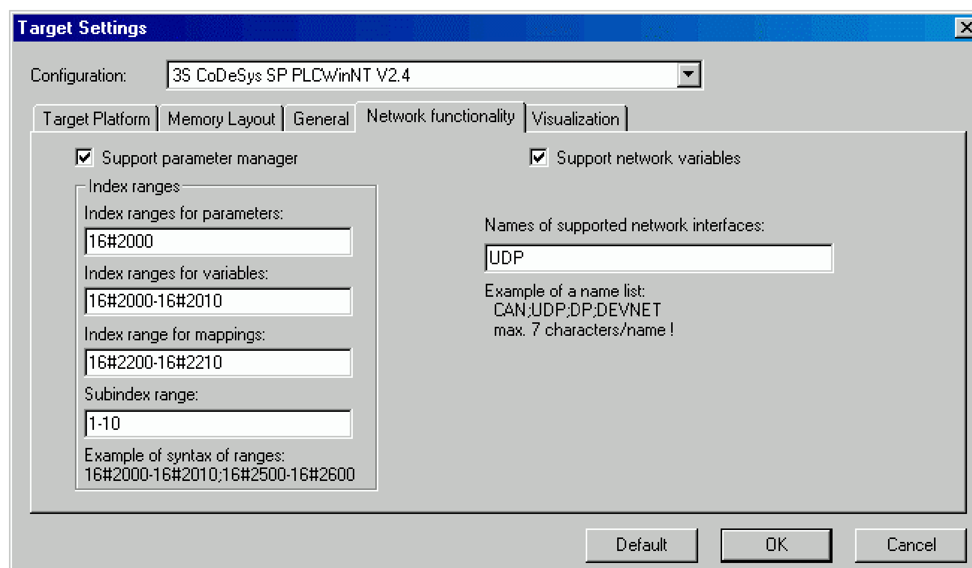


Figura 6-21. Ativando o Gerenciador de parâmetros

Características do editor de gerenciamento de parâmetros

Na guia Recursos escolha o objeto 'Gerenciador de parâmetros'. Uma janela do Editor será aberta, onde pode-se criar, editar e armazenar listas de parâmetros e, no modo online, também pode carregá-las para o sistema do dispositivo e monitorar os valores de parâmetros atuais.

Nota: para ter disponível a funcionalidade do Gerenciador de Parâmetros em um projeto MasterTool IEC, a opção 'Suporte ao gerenciador de parâmetros' nas configurações do dispositivo deve estar ativada e intervalos de índice apropriados devem ser definidos lá!

A janela Editor é bipartida. A parte esquerda é usada para navegação, e exibe uma lista de todas as listas de parâmetros carregadas no momento para o Gerenciador de Parâmetros. A parte direita contém um editor de tabela, com os títulos das colunas contendo os nomes dos atributos.

A **janela de navegação** permite inserir, excluir, reorganizra ou renomear listas de parâmetros de tipos diferentes (Variáveis, Parâmetros, Constante, Modelo, Instância, Parâmetros do Sistema).

No **editor de tabela** preenche-se a lista com entradas de parâmetros. Cada tipo de lista mostrará uma seleção especial de colunas de atributo que podem ser editadas ou apenas estão visíveis. Ou isso é definido por um arquivo de descrição específico do dispositivo ou as configurações padrão serão consideradas.

Pode-se saltar entre janela de navegação e a tabela do editor pressionando <F6>.

Editor do Gerenciador de parâmetros do MasterTool IEC:

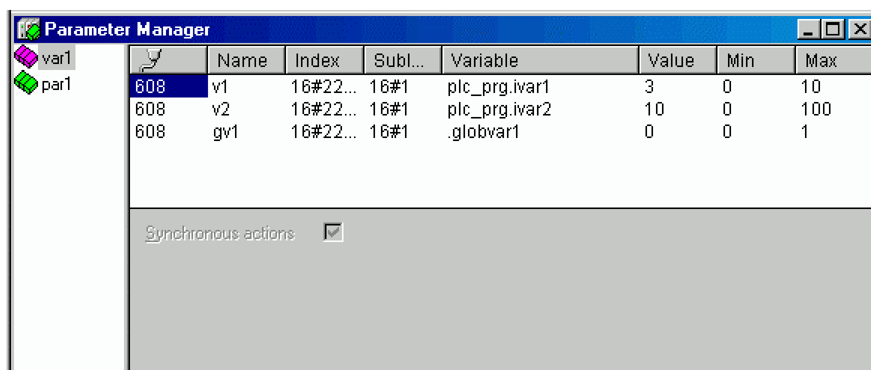


Figura 6-22. Editor do Gerenciador de parâmetros do MasterTool IEC

No modo online é possível carregar as listas, criadas anteriormente, para o sistema do dispositivo conectado no momento. Além disso, pode-se usar as funções do Gerenciador de Parâmetros para acessá-las para fins de troca de dados com outros sistemas (escrever valores, carregar). Mais tarde na janela do Gerenciador de Parâmetros, pode-se monitorar os valores atuais dos parâmetros. Se, no momento, nenhuma conexão online é estabelecida, as listas de parâmetros apenas podem ser criadas localmente e salvas com o projeto.

Tipos e atributos da lista de parâmetros

O Gerenciador de Parâmetros pode manipular os seguintes tipos de lista de parâmetros:

- **Variáveis**: as entradas em listas de parâmetros desse tipo representam variáveis do processo no projeto.
- **Parâmetros**: as entradas em listas de parâmetros desse tipo representam parâmetros cujos valores não estão atrelados ao processo.
- **Parâmetros do sistema**: as entradas em listas de parâmetros desse tipo representam dados que não estão atrelados ao processo e que são determinados pelo sistema do dispositivo. Listas de Parâmetros do sistema não podem ser excluídas nem renomeadas.
- **Modelo**: um modelo não contém entradas de parâmetro que podem ser acessadas diretamente com a finalidade de troca de dados. Na verdade, as entradas fornecem uma "configuração de atributo básico" para os componentes de um bloco funcional ou uma estrutura. Assim, um modelo pode ser usado nas listas de parâmetros do tipo 'Instância'.
- **Instância**: as entradas em listas de parâmetros deste tipo representam entradas de parâmetro para variáveis do tipo bloco funcional ou estrutura, o que significa que são instâncias ou estruturas variáveis. Para uma fácil inserção dos parâmetros, é usado um modelo, o qual já está criado no Gerenciador de Parâmetros.
- **Mapeamentos**: este tipo de lista só está disponível, se suportado pelo sistema do dispositivo. As entradas representam variáveis do processo que devem ser usadas no mapeamento PDO de um Dispositivo CAN. Portanto, listas de mapeamento basicamente são listas "Variáveis", porém operando em um intervalo separado de índice/sub-índice. Esse intervalo deve ser definido nas configurações do dispositivo, categoria a Funcionalidade de rede! Nesse caso um Dispositivo CAN, que é configurado na configuração do CP, só usará as entradas da Lista tipo 'Mapeamento', enquanto, por outro lado, todas as entradas das listas de variáveis ou instâncias estarão disponíveis na caixa de diálogo de mapeamento PDO.

Cada tipo de lista será exibido no editor do Gerenciador de Parâmetros de acordo com os atributos definidos pelo sistema do dispositivo.

Instâncias e Modelos

Uma lista de parâmetros “Instância"...

... trata as entradas de parâmetro, que representam um **bloco funcional**, uma **estrutura** variável ou uma **matriz**. Listas de instância para um bloco funcional ou uma estrutura são baseadas em um **modelo** que também deve ser definido no Gerenciador de Parâmetros para a estrutura e o bloco

funcional respectivos. Listas de instância para matrizes não podem usar um modelo feito no Gerenciador de Parâmetros, mas diretamente se referem à matriz que é usada no projeto.

Uma lista de parâmetros "Modelo"...

... não contém parâmetros que são acessados diretamente com a finalidade de troca de dados. Na verdade ela define os offsets de índice e sub-índice e determinados atributos para as entradas de parâmetros que representam os componentes de um bloco funcional ou de uma estrutura. O modelo pode ser usado em uma lista de parâmetros 'Instância' (consultar acima), assim fornecendo uma maneira fácil de criar entradas de parâmetros para variáveis do projeto que são instâncias de um bloco funcional ou uma estrutura.

Criando uma lista de parâmetros modelo:

No campo de edição próxima a **base da POU** insira o nome do bloco funcional ou estrutura para o qual um modelo de parâmetro deve ser criado. Usando o assistente de entrada pode-se procurar os POUs disponíveis do projeto. Pressione '**Aplicar'** para inserir os componentes da POU escolhida o Editor de lista de parâmetros. Agora edite os campos de atributo e feche a lista para torná-la disponível para uso em uma lista 'Instância'.

Se a opção **Ações síncronas** está ativada, todos os acessos de leitura/escrita em outras POUs definidas para as entradas da lista serão executados pelo sistema do dispositivo em sincronia com a chamada da entrada respectiva.

O comando para **inserir entradas faltantes** no menu de contexto ou no menu 'Extras' fará uma atualização das entradas de acordo com a versão atual da POU base. Isso pode ser necessário após ter excluído algumas linhas ou após ter alterado a POU base.

Para criar listas de parâmetros de instância para **matrizes** não é necessário criar um modelo no Gerenciador de Parâmetros. O modelo ARRAY estará disponível implicitamente.

Criando uma lista de parâmetros de instância:

Edite um **Modelo** a partir da lista de seleção abaixo da tabela. Esta lista oferece todos os modelos disponíveis no momento para blocos funcionais ou estruturas no Gerenciador de Parâmetros juntamente com a opção ARRAY, o qual é selecionada caso deseje-se referir diretamente a uma matriz usada no projeto. Pressione '**Aplicar'** para inserir os componentes predefinidos para a tabela da lista de parâmetros.

Para opção **Ações Síncronas** consultar acima: Criando uma Lista de parâmetros modelo.

No campo de edição **Variável base** digite exatamente que variável do projeto (deve ser do tipo do bloco funcional ou da estrutura ou da matriz que é descrita pelo modelo escolhido), para os componentes dos quais deseja-se criar entradas de parâmetros.

Insira um índice base e sub-índice base para a instância. Os índices e sub-índices dos componentes específicos, então, serão calculados automaticamente adicionando os valores de índice e sub-índice que estão definidos no modelo para cada componente (no caso de matrizes a base será 0). Eles serão preenchidos automaticamente para os campos de atributo respectivos. Exemplo: ao inserir um índice base "3" para um componente, para o qual um índice de offset "3000" é definido no modelo, o componente será ajustado com o índice 3003.

O comando **'Inserir entradas faltantes'** no menu de contexto ou no menu 'Extras' fará uma atualização das entradas de acordo com a versão atual do modelo usado. Isso pode ser útil após entradas terem sido excluídas ou após modificação no modelo.

Exemplo

Crie um bloco funcional fubo com variáveis de entrada ou saída: a, b, c. No PLC_PRG defina as seguintes instâncias FB: inst1_fubo:fubo; inst2_fubo:fubo. Compile o projeto.

Agora abra o Gerenciador de parâmetros a fim de criar lista de parâmetros para variáveis inst1_fubo.a, inst1_fubo.b, inst1_fubo.c e inst2_fubo.a, inst2_fubo.b, inst2_fubo. Primeiro insira uma lista de parâmetros do tipo 'Modelo' e nomeie-a "fubo_template". Defina a POU base: "fubo". Pressione Aplicar e defina alguns atributos para os componentes a, b, c: te. Insira os deslocamentos

de índice: para a: 16# 1, para b: 16# 2, para c: 16# 3. Também o sub-índice compensa, por exemplo, a: 16# 2, b: 16# 3, c: 16# 4.

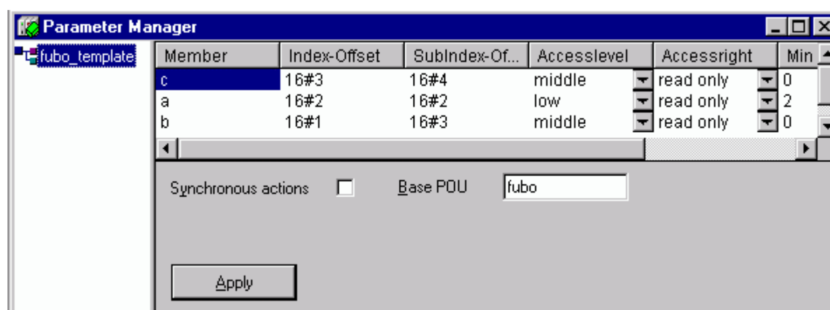


Figura 6-23. Criando uma lista de parâmetros de instância (parte 1)

Agora, insira uma nova lista de parâmetro do tipo «instância». Escolha modelo "fubo_template". Insira a variável de base "inst1_fubo". Defina um índice de base d, por exemplo, 16# 2300 e uma base (sub-índice) de 30 (deve-se considerar os intervalos definidos nas configurações do dispositivo na guia Funcionalidade de rede!). Agora pressione Aplicar para exibir os índices que são calculados para os componentes a, b, c pela adição de deslocamentos de base e deslocamentos de modelo definido: índices: 16# 2301, 16# 2302, 16# 2303; Sub-índices: 16#23, 16#33, 16#43.

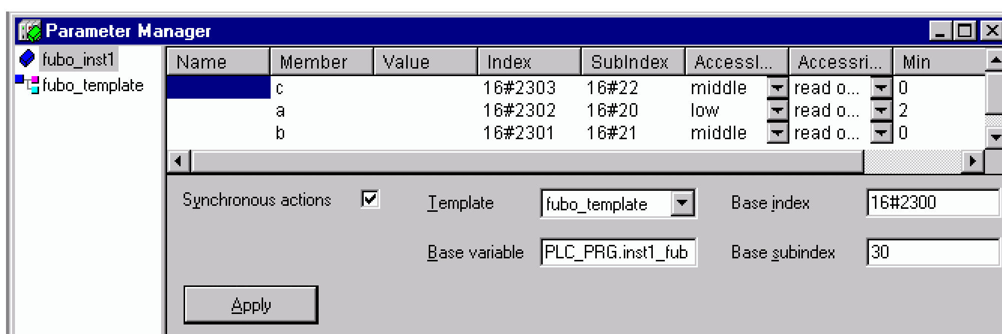


Figura 6-24. Criando uma lista de parâmetros de instância (parte 2)

Baseado nestas entradas automaticamente criadas, agora pode-se continuar a editar a lista de parâmetros.

Gerenciando listas de parâmetros

Inserir Lista

Atalho: Ins \

Para inserir uma nova lista de parâmetros use o comando 'Inserir lista' ou 'Inserir nova lista' no menu 'Inserir' acessível também pelo menu de contexto. Os comandos estão disponíveis quando o foco está na janela de navegação vazia e em uma entrada já existente na árvore de navegação.

Insira um Nome para a nova lista de parâmetros (deve ser exclusivo no tipo de lista) e escolha um dos tipos de lista a seguir:

- Variáveis — Entradas para variáveis do processo
- Parâmetros — Entradas de dados, cujos valores não estão vinculados ao processo
- Modelo — Modelo de configuração do atributo para os componentes de um bloco funcional ou uma estrutura

- Instância — Entradas para as variáveis do tipo bloco funcional ou estrutura, baseada no modelo correspondente (veja acima)
- Mapeamentos — Entradas para variáveis do processo, a serem usadas no mapeamento PDO de um dispositivo CAN. Esse tipo só está disponível, se suportado pelo sistema do dispositivo!
- Parâmetros do sistema — Entradas de parâmetros cujos valores não estão conectados pelo processo e que são definidos pelo sistema do dispositivo.

Diálogo 'Inserir lista':

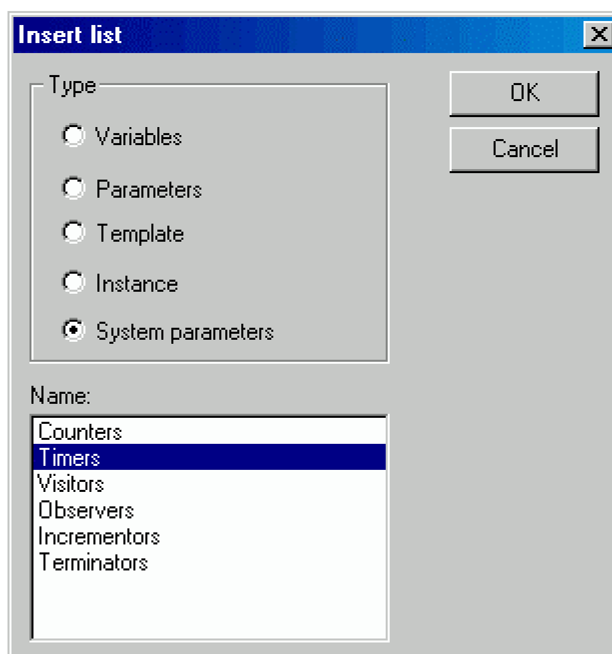


Figura 6-25. Diálogo 'Inserir lista'

Após confirmar as configurações e fechar a caixa de diálogo com OK, a nova lista de parâmetros irá aparecer como uma entrada na janela de navegação, sendo o tipo de lista indicado pelo ícone. No Editor de tabela os atributos apropriados serão exibidos como títulos de colunas. A seleção e ordem das colunas são definidas por um arquivo de descrição específico do dispositivo, caso contrário, as configurações padrão serão usadas. Agora pode-se editar a tabela, inserindo uma linha para cada parâmetro desejado (veja neste mesmo capítulo Editando listas de parâmetros).

Renomear lista

A lista de parâmetros, que atualmente esteja marcada na janela de navegação, pode ser renomeada usando o comando 'Renomear lista' que está disponível no menu 'Extras' ou no menu de contexto. Um campo de edição será aberto, este campo também pode ser aberto através de umclique duplo com o mouse sobre o nome da lista.

Recortar/Copiar/Colar lista

Atalhos: <Ctrl>+ <X>, <Ctrl>+ <C>, <Ctrl> +<V>,

O comando 'Recortar' (menu Editar) e 'Recortar lista' (menu de contexto) move a lista selecionada da janela de navegação para um buffer temporário, de modo que é possível reinseri-lo utilizando o comando 'Colar' em qualquer outra posição na árvore de navegação. Antes de colar, selecione a posição na qual será inserida a lista recortada.

O comando 'Copiar' e 'Copiar lista' também usa o buffer temporário, mas a árvore de navegação original será mantida e uma cópia será adicionada via 'Colar'.

Apagar lista

Atalho: <Del>

A lista selecionada atualmente na janela de navegação será removida pelo comando 'Excluir' (menu 'Editar') e 'Excluir lista' ('Extras') , acessível também pelo menu de contexto

Atenção: no modo online esse comando excluirá a lista correspondente no controlador.

Editando listas de parâmetros

Quais colunas (atributos) são exibidas?

A lista de parâmetros atualmente marcada será exibida na janela que exibe a tabela, conforme definido pelo arquivo de descrição específico do dispositivo e de acordo com a configuração padrão.

Isso significa que os valores dos atributos de cada parâmetro específico serão exibidos em uma **linha** separada de acordo com a ordem específica do tipo de lista e seleção de colunas.

Pode-se **ocultar** ou **exibir** colunas através da desativação/ativação das mesmas através do menu de contexto quando o cursor está apontando para qualquer campo da barra de título da coluna da lista.

Para modificar a coluna mova os divisores entre os campos de título de coluna ou use um dos comandos disponíveis no menu de contexto quando o cursor é colocado em um campo de título de coluna: o comando **Largura da coluna padrão** definirá uma largura padrão para todas as colunas. **Maximizar a largura** alterará a largura da coluna focalizada no momento para que cada entrada seja exibida totalmente.

Comandos para editar uma entrada da lista de parâmetros:

Os comandos a seguir servem para a edição de uma lista de parâmetros e estão disponíveis no **menu de contexto** e nos menus **'Inserir'** ou **'Extras'**:

Inserindo/Excluindo linhas:

Inserir linha e Nova linha	Uma nova entrada (linha) será inserida antes da posição atual do cursor.
Linha Após e Nova linha após **Atalho: <Ctrl> <Enter>**	Uma nova entrada (linha) será inserida depois da posição atual do cursor.
Excluir linha **Atalho para excluir linha: <Shift>+<Del>**	A linha, onde o cursor estiver colocado será excluída.
Recortar, Copiar, Colar Linha	Esses comandos podem ser usados para mover (Recortar/Colar) ou para copiar as linhas (copiar/colar) selecionadas.

Tabela 6-1. Inserindo/Excluindo linhas

Editando valores de atributo:

Se uma nova linha para uma entrada de parâmetro é inserida, os campos de atributo serão automaticamente preenchidos com valores padrão específicos do dispositivo. Para inserir ou editar um valor de atributo, clique no campo correspondente. Um campo de edição será aberto, se o atributo for editável. O assistente de entrada (<**F2**>) estará disponível nos campos onde um componente do projeto MasterTool IEC deve ser inserido.

Pressione <**Enter**> para fechar a entrada.

Usando as teclas de seta pode-se saltar para outro campo.

Pressione <**Del**> para excluir a entrada do campo editado no momento.

Para alternar o formato de entrada entre 'decimal' e 'hexadecimal' use o comando **'Formatar Dec/Hex'** no menu **'Extras'**.

Pressione <**F6**> para definir o foco para a janela de navegação.

Opções:

Abaixo da tabela, na janela do editor, as seguinte opções podem ser ativadas (a disponibilidade dependendo do tipo de lista):

Download com programa: em um download do programa a lista será enviada automaticamente para o controlador.

Ações síncronas: todos os acessos de leitura/escritas em outras POUs definidas para quaisquer entradas da lista serão executados pelo sistema do dispositivo em sincronia com a chamada da entrada respectiva.

Classificando listas de parâmetros

A seqüência de entradas em uma lista de parâmetros pode ser classificada em relação a um atributo (coluna) na ordem dos valores de atributo crescente ou decrescente. Isso funciona no modo offline e online.

Execute um clique no mouse no campo que contém o título da coluna do atributo desejado. Assim, as linhas da tabela serão classificadas novamente e no campo de título da coluna do atributo, um símbolo de seta será exibido, mostrando a classificação atual (apontando para cima = seqüência de classificação crescente, apontando para baixo = seqüência de classificação decrescente).

Gerenciador de parâmetros no modo online

Transferência de listas entre o Editor e Unidade Controladora

Se suportado pelo dispositivo, no modo online as listas de parâmetros, que foram criadas no Editor, podem ser **enviadas** e **lidas** a partir do controlador. O tamanho máximo de listas do tipo 'Variável' e 'Parâmetros' também é definido pelo sistema do dispositivo.

Atenção: Durante o login, o download de todas as listas de parâmetros será feito se a opção 'Carregar com projeto' estiver ativada!

O valor atual de cada parâmetro é **monitorado** em uma coluna adicional que é exibida no Gerenciador de Parâmetros no modo online:



Figura 6-26. Monitoração do valor atual de cada parâmetro

O monitoramento dos valores pode ser feito através do Índice e Sub-índice ou do RefID e Offset, dependendo do dispositivo.

Os comandos a seguir servem para tratar a tranferência de listas entre o editor e o controlador, estes comandos estão disponíveis no menu 'Extras':

Excluir lista	A lista selecionada na janela de navegação será excluída do controlador.
Escrever lista	Esse comando abrirá a caixa de diálogo 'Copiar objetos' onde pode-se selecionar, entre as listas disponíveis, aquelas que deseja-se fazer o download para o controlador. O download será feito logo após confirmação com 'OK'. Depende do dispositivo se, para enumerações, somente os valores numéricos ou, adicionalmente, valores simbólicos serão transferidos.
Lista de leitura	Todas as listas do tipo 'Parâmetros' serão lidas a partir do controlador e carregadas no Gerenciador de Parâmetros. A "leitura" das listas do tipo 'variáveis' será feita somente se for suportado pelo dispositivo.
Escrever valores	Todos os valores definidos na coluna 'Valor' serão gravados na lista de parâmetros no controlador. Para escrever valores únicos, execute um clique duplo no respectivo campo da coluna , assim abrindo a caixa de diálogo ‘Escrever valor’, da mesma forma como é feito na função ‘Comunicação’ 'Escrever valores'.
Escrever valores padrão	Os valores definidos na coluna 'Padrão' serão gravados na lista de parâmetros no controlador.
Assumir valores	Os valores atuais serão lidos do controlador e serão carregados para a coluna 'Valor'.

Tabela 6-2. Comandos para a transferência de listas entre o editor e controlador

O comando **Formatar Dec/Hex** também está disponível para alternar o formato de entrada entre 'Decimal' e ‘Hexadecimal’ nas listas de parâmetros em um projeto de inicialização.

Listas de parâmetros em um projeto de inicialização

Dependendo do sistema do dispositivo, as listas de parâmetro serão consideradas quando um projeto de inicialização é criado.

Exportação e importação de listas de parâmetros

'Extras' 'Exportar'

O comando 'Exportar' no menu 'Extras' pode ser usado para exportar a lista do Gerenciador de Parâmetros para um arquivo XML. Esse arquivo, por exemplo, pode ser importado para outro projeto usando a função importar do gerenciador de parâmetros do MasterTool IEC. A caixa de diálogo padrão para salvar um arquivo será aberta, nela a extensão de arquivo *.prm estará pré-definida. Todas as listas disponíveis no Gerenciador de Parâmetros serão gravadas para o arquivo de exportação.

O conteúdo do Gerenciador de Parâmetros também pode ser exportado usando a função de exportação do projeto geral ('Projeto' 'Exportar').

'Extras' 'Importar'

O comando 'Importar' do menu 'Extras' pode ser usado para importar um arquivo XML que descreve as listas de parâmetros. Esse arquivo, por exemplo, pode ser criado usando a função exportar no Gerenciador de Parâmetro do MasterTool IEC.

Se o arquivo de importação contém uma lista já nomeada no Gerenciador de Parâmetros, será aberta uma caixa de diálogo perguntando ao usuário se a lista existente deve ser substituída.

Configurações do dispositivo

As "Configurações do dispositivo" são um objeto de 'Recursos'. Aqui é definido, qual dispositivo deve ser usado para o projeto e como ele será configurado. Se um novo projeto é iniciado através do menu **'Projeto' 'Novo'**, uma caixa de diálogo será aberta onde deve ser definida uma pré-configuração para um dispositivo.

A lista dos dispositivos disponíveis depende de quais Pacotes de Suporte do Dispositivo (TSP) estão instalados no computador. Eles descrevem configurações básicas específicas de uma plataforma e também definem o quanto a configuração pode ser modificada pelo usuário nas caixas de diálogo de configurações do dispositivo no MasterTool IEC.

Atenção: se não houver TSP disponível, somente a configuração 'Nenhum' será oferecida na caixa de seleção do sistema do dispositivo. Isso irá automaticamente alternar para o modo de simulação e nenhuma definição de configuração será possível.

Pacote de suporte do dispositivo

Um Pacote de Suporte do Dispositivo (TSP) deve ser instalado antes de iniciar a ajuda do Programa de Instalação **InstallTarget** que é parte da configuração do **MasterTool IEC**.

Um Pacote de Suporte do Dispositivo (**TSP**) contém todos os arquivos e informações de configuração necessárias para controlar uma plataforma padrão com um programa criado no **MasterTool IEC**. O que deve ser configurado: gerador de código, leiaute de memória, funcionalidade do CP, módulos de E/S. Além disso, bibliotecas, drivers de gateway, arquivos ini para mensagens de erro e Navegador-CP devem ser vinculados. O componente central de um TSP é um ou mais Arquivos do Dispositivo. Um Arquivo do Dispositivo direciona para arquivos adicionais, os quais são necessários para configurar o dispositivo. É possível que vários arquivos do dispositivo compartilhem esses arquivos adicionais.

A extensão padrão para um arquivo do dispositivo é ***.trg**, seu formato é binário. Definições adicionais são anexadas às entradas no arquivo do dispositivo que determinam se o usuário pode ver e editar as configurações nas caixas de diálogo do **MasterTool IEC**.

Durante a instalação de um TSP o arquivo do dispositivo para cada dispositivo é colocado em um diretório separado e o caminho é registrado. Os arquivos associados são copiados para o computador de acordo com as informações de um Arquivo de **informações *.tnf**. O nome do diretório do dispositivo é o mesmo nome do dispositivo. É recomendável armazenar os arquivos específicos do dispositivo em um diretório que seja nomeado com o nome do fabricante.

Os arquivos que são instalados com um TSP são lidos quando o **MasterTool IEC** é iniciado. As configurações do dispositivo que são feitas nas caixas de diálogo do **MasterTool IEC** serão salvas com o projeto.

Nota: ao usar um novo arquivo do dispositivo ou alterar o já existente, o **MasterTool IEC** tem que ser reiniciado para ler a versão atualizada.

Diálogo de configurações do dispositivo

O diálogo Configurações do dispositivo será aberto automaticamente, quando um novo projeto é criado. Ele também pode ser aberto selecionando o item de menu **'Configurações do dispositivo'** no registro 'Recursos' na biblioteca do Organizador de objetos.

Escolha uma das configurações do dispositivo oferecidas na Configuração.

Se nenhum pacote de suporte ao dispositivo tiver sido instalado, apenas **'Nenhum'** pode ser selecionado, o que significa trabalhar no modo de simulação. Ao escolher uma das configurações instaladas , as possibilidades que são deixadas para personalizar essa configuração nas caixas de diálogo do **MasterTool IEC** dependem. Ao escolher uma configuração do dispositivo, para a qual não existe nenhuma licença válida no computador, o **MasterTool IEC** solicita a escolha de outro dispositivo.

Se uma configuração fornecida com a entrada 'Ocultar configurações' é selecionada no arquivo do dispositivo correspondente, somente poderá ser visualizado o nome da configuração. Caso contrário, há cinco caixas de diálogo disponíveis para modificar a configuração dada:

1. Plataforma do dispositivo;
2. Leiaute de memória;
3. Geral;
4. Funcionalidade de rede;
5. Visualização.

Atenção: esteja ciente de que cada modificação da configuração do dispositivo pré-definido pode causar alterações graves no desempenho e no comportamento do dispositivo!

Pressione <Default> para redefinir as configurações do dispositivo para a configuração padrão fornecida pelo arquivo do dispositivo.

O Navegador CP

O Navegador CP é um monitor de controle baseado em texto (terminal). Comandos para a solicitação de informações específicas a partir do controlador são inseridos em uma linha de entrada e enviados como seqüência de caracteres para o controlador. A seqüência de caracteres de resposta retornada é exibida em uma janela de resultados do navegador. Essa funcionalidade serve de diagnóstico e para fins de depuração.

Os comandos disponíveis para o sistema do dispositivo são compostos pelo conjunto padrão do MasterTool IEC além de um possível conjunto de extensão do fabricante do controlador. Eles são gerenciados em um arquivo INI e implementados de acordo no controlador.

Considerações gerais referentes à operação do Navegador CP

Selecione a entrada Navegador CP na guia de controle Recursos. Ela estará disponível lá se estiver ativada nas configurações do dispositivo atual (categoria Funcionalidade de rede).

A janela do Navegador CP MasterTool IEC:

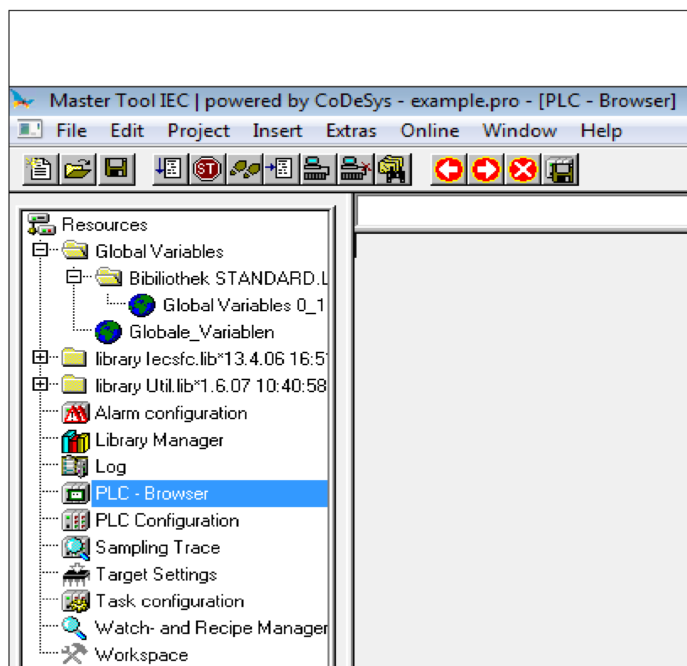


Figura 6-27. A janela do Navegador CP MasterTool IEC

O navegador consiste em uma linha de comando de entrada e uma janela de resultado/exibição

Em uma caixa de seleção, a linha de entrada exibe uma lista de todos os comandos inseridos desde o início do projeto (histórico de entrada). Eles estão disponíveis para re-seleção até que o projeto seja fechado. Somente comandos, que diferem daqueles já existentes, são adicionados à lista.

O comando digitado é enviado para o controlador com <Enter>. Se não há nenhuma conexão online, o comando é exibido na janela da mesma forma que quando eles são enviados para o controlador, caso contrário, a resposta do controlador é mostrada lá. Se um novo comando for enviado para o controlador, o conteúdo da janela de resultados é excluído.

Os comandos podem ser inseridos no formulário de strings de comando, o uso de macros é possível também.

Comandos de entrada no Navegador CP

Basicamente o Navegador CP torna disponíveis os **comandos padrão ALTUS** embutidos no controlador. Ele diz respeito a funções para a manipulação direta de memória, para a saída do projeto e status de funções, bem como para o monitoramento no controlador. Eles são descritos na **INI** do arquivo navegador, que é parte integrante do pacote de suporte do dispositivo. Esses comandos padrão podem ser complementados posteriormente por aqueles especializados, por exemplo, funções de diagnóstico próprias ou outras mensagens de status do aplicativo de controle. A expansão da lista de comando deve ser realizada na interface do cliente no controlador bem como por meio de entradas adicionais no arquivo navegador INI.

Ao abrir o projeto, a **lista de comandos** disponíveis no Navegador CP é gerada com base nas entradas no arquivo navegador INI. Ele pode ser acessado como entrada de ajuda usando o botão aproprido na caixa de diálogo “Inserir comando padrão” ou usando <F2>. Também o comando **'Inserir' 'Comandos padrão'** pode ser usado para obter a lista de comandos. Um comando pode ser digitado manualmente na linha de comando ou ele pode ser selecionado na lista por um duplo clique na entrada apropriada.

A sintaxe de comando é:

<KEYWORD><LEER><KEYWORD-DEPENDEND PARAMETERS>

A keyword é o **comando**. Os **parâmetros** que podem ser expandidos estão descritos na tooltip respectiva na janela de ajuda.

O comando que foi enviado é repetido na janela de dados de saída, a resposta do controlador aparece abaixo.

Exemplo: solicitação para o Id de projeto do controlador com o comando "PID"

Entrada na linha de comando:

PID

Saída na janela de resultados:

PID

PROJECT-ID: 16#0025CFDA

Um **texto de ajuda** pode ser fornecido para cada comando padrão com?<BLANK><KEYWORD>. De forma similar, isso está definido no arquivo INI.

Os comandos a seguir estão integrados no controlador e contidos o arquivo INI com as entradas correspondentes para entrada de ajuda, tooltip e ajuda:

Comando	Descrição
?	O controlador fornece uma lista dos comandos disponíveis. A lista é independente do status dos arquivos de descrição do sistema do dispositivo.
mem	Hexdump de um espaço de memória Sintaxe 1: mem <endereço inicial> <endereço final> Endereços podem ser inseridos em decimal, hexadecimal (prefixo 16#) ou como uma macro.
memc	Hexdump relativo ao endereço inicial do código no controlador; como em mem, os dados são adicionados ao endereço inicial da área de código.
memd	Hexdump relativo ao endereço da base de dados no controlador; como em mem, os dados são adicionados ao endereço inicial da área de código.
reflect	Reflete a linha de comando atual, para fins de teste.
dpt	Lê e exibe a tabela de ponteiro de dados.
ppt	Lê e exibe a tabela POU.
pid	Lê e exibe a Id do projeto.
pinf	Lê e exibe informações de projeto (consultar 'Projeto' 'Informações de projeto').
tsk	Mostra a lista de tarefas IEC incluindo informações de tarefa definidas no projeto.
startprg	Start do programa do CP (‘Comunicação’ 'Run')
stopprg	Stop do programa do CP (‘Comunicação’ 'Stop')
resetprg	Reset do programa do CP. Apenas os dados não-retentivos serão inicializados. (‘Comunicação’ 'Reset’).
resetprgcold	Reset (a frio) do programa do CP. Dados retentivos também são inicializados. (‘Comunicação’ 'Reset (a frio)').
resetprgorg	Reset (original) do programa do CP. O programa aplicativo atual, bem como todos os dados (incluindo retentivos e persistentes) são excluídos. (‘Comunicação’ 'Reset (original)').
reload	Recarrega o projeto de inicialização.
getprgprop	Lê e exibe as propriedades do programa (nome, título, autor, versão e data).
getprgstat	Lê e exibe o status do programa (por exemplo, "run", "stop", último erro, sinalizadores)
filedir	Comando de arquivo "dir". Lista de arquivos no diretório do CP.
filecopy	Copia o arquivo [de] [para]. Exemplo: "filecopy filename.txt filename2.txt".

filerename	Renomeia arquivos no CP [antigo] [novo]. Exemplo: “filerename oldname.txt newname.txt"
filedelete	Exclui arquivos no CP; exemplo: "filedelete file.xml".
saveretain	Salva variáveis retentivas. O nome do arquivo de salvamento será exibido posteriormente.
restoreretain	Carrega variáveis retentivas. O nome do arquivo de salvamento, a partir do qual os valores de variáveis são restaurados, será exibido.
setpwd	Define a senha no controlador; sintaxe: setpwd <senha> [nível], por exemplo, "SETPWD abcde 0" <nível> pode ser "0" (padrão) apenas válido em logins do sistema de programação, ou "1" ,válido para todas as aplicações
delpwd	Apaga senha no CP.

Tabela 6-3. Comandos de entrada no Navegador CP

Considerar:

- A primeira palavra da seqüência de comando digitada é interpretada como **palavra-chave** (<KEYWORD>).
- Se a primeira palavra da entrada do comando **não é reconhecida** pelo controlador, a resposta 'Palavra-chave não encontrada' será exibida na janela de resultados.
- Se uma palavra-chave é precedida por um **„? <SPACE> "**(ex. „? mem"), o arquivo INI será pesquisado quanto à existência de uma seção de ajuda para esta palavra-chave. Se estiver disponível, nada é enviado para o controlador, mas apenas o texto de Ajuda é exibido na janela de dados de saída

Uso de macros durante o comando de entrada no Navegador CP

Se um comando associado a uma macro é inserido na linha de comando, isso será expandido antes de ser enviado para o controlador. A resposta na janela Resultado aparecerá em um formulário expandido na mesma maneira.

A sintaxe de entrada é: <KEYWORD><macro>

<KEYWORD> é o comando.

As macros são:

%P<NAME>	Se NAME for um nome de POU, a expressão é expandida para <POU-Index>, caso contrário, não haverá nenhuma alteração
%V<NAME>	Se NAME for um nome de variável, a expressão é expandida para #<INDEX>:<OFFSET>, caso contrário, não há nenhuma alteração (esta notação #<INDEX>:<OFFSET> é interpretada pelo controlador como um endereço de memória)
%T<NAME>	Se NAME for um nome de variável, a expressão é expandida para <VARIABLENTYP>, caso contrário, não há nenhuma alteração.
%S<NAME>	Se NAME for um nome de variável, a expressão é expandida para <SIZEOF(VAR)>, caso contrário, não há nenhuma alteração.

Tabela 6-4. Macros no comando de entrada do Navegador CP

O caractere % será ignorado se o símbolo \ é colocado na frente. O símbolo \ somente será transmitido se escrito \\.

Exemplo:

Entrada na linha de comando: (Dump de memória da Variável .testit ?)

```
MEM %V.TESTIT
```

Saída na janela de resultado:

```
MEM #4:52
03BAAA24 00 00 00 00 CD CD CD CD ....ÍÍÍÍ
```

Opções adicionais do Navegador CP

No menu **'Extras'** ou na barra de ferramentas do Navegador do CP existem os seguintes comandos para manipular a entrada do comando ou a lista de histórico:

Com os **comandos Avançar histórico** e **Retornar histórico** pode-se rolar para trás e para frente nas ocorrências já geradas. A gravação do histórico continua até que o projeto seja fechado.

Com o **comando 'Cancelar'** pode-se suspender uma query que tenha sido iniciada.

Como o **comando Salvar lista de histórico** pode-se salvar os resultados de query executados até um determinado ponto em um arquivo texto externo. A caixa de diálogo 'Salvar arquivo como' será exibida, onde pode-se digitar um nome de arquivo com a extensão „.bhl" (Browser History List). O comando **Imprimir último comando** abre a caixa de diálogo padrão para imprimir. O query atual mais os dados de saída na janela da mensagem podem ser impressos.

Ferramentas

O objeto 'Ferramentas' estará disponível na guia 'Recursos' se a funcionalidade é ativada para o sistema do dispositivo atualmente definido. Ela mostra todos os atalhos disponíveis (conexões) para arquivos executáveis de ferramentas externas, que podem ser ativados por um clique duplo para chamar esses programas externos a partir do MasterTool IEC. Estão definidos pelo arquivo do dispositivo quais e quantos atalhos são permitidos. Dependendo esta definição o usuário pode adicionar ou excluir novos atalhos na pasta 'Ferramentas'.

Por exemplo, a pasta Ferramentas no Organizador de objetos poderia ter esta aparência:

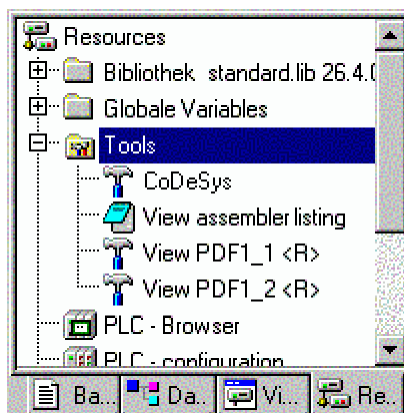


Figura 6-28. Ferramentas no Organizador de objetos

Nesse exemplo quatro atalhos de ferramentas estão instalados. Um serve para iniciar outro sistema de programação MasterTool IEC, um para abrir a listagem Assembler em um editor de texto e os outros dois atalhos adicionais estão disponíveis para abrir arquivos PDF. Atalhos marcados com um "<r>" não podem ser modificados no MasterTool IEC. Os atalhos podem conter a conexão com um editor, por exemplo, notepad.exe ou para um determinado arquivo PDF, de forma que um clique duplo na entrada possa abrir uma janela do Bloco de Notas mostrando a listagem Assembler ou abrir o Acrobat Reader para mostrar o arquivo PDF respectivamente.

Além disso, pode-se definir determinados arquivos que devem ser enviados para o CP assim que o atalho for ativado.

Propriedades dos atalhos de ferramentas disponíveis (propriedades de objeto)

Via clique do mouse no sinal de mais na entrada 'Ferramentas', na guia Recursos do Organizador, uma lista dos atalhos disponíveis será aberta. Se apenas a configuração de um novo projeto estiver sendo iniciada, somente aparecerão aquelas que estão definidas no arquivo do dispositivo como entradas fixas. Mas, se a pasta Ferramentas já tiver sido modificada, poder-se-ão encontrar outros atalhos, adicionados por um usuário no MasterTool IEC.

Caixa de diálogo 'Propriedades da ferramenta':

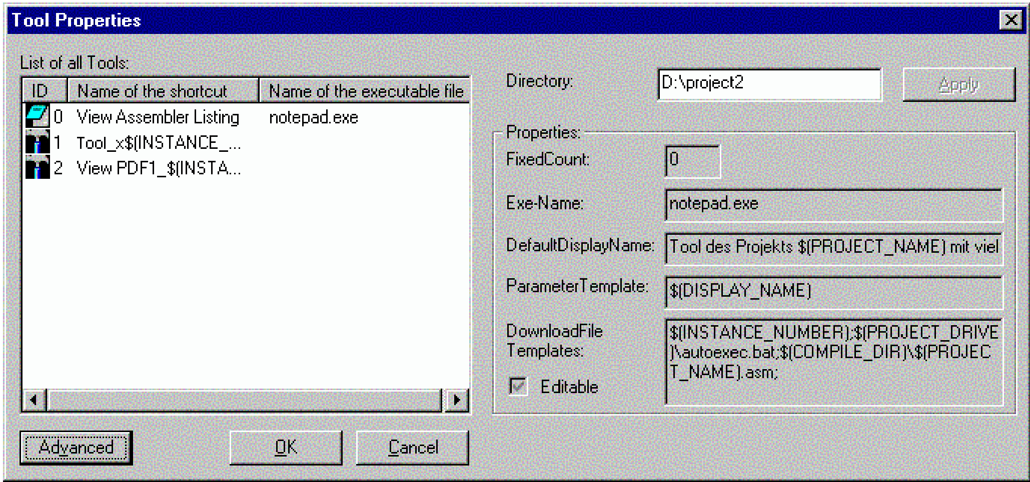


Figura 6-29. Caixa de diálogo 'Propriedades da ferramenta'

Pode-se exibir as propriedades das ferramentas globais (válidas para todos os atalhos listados em 'Ferramentas'), bem como as propriedades de atalhos únicos.

1. Propriedades da ferramenta

Se 'Ferramentas' estiver marcada na árvore de Recursos, estará disponível o comando 'Propriedades do Objeto' no menu de contexto ou no menu 'Projeto' 'Objeto', que abrirá a caixa de diálogo 'Propriedades da Ferramenta'.

Lá obtém-se uma tabela listando todos os atalhos de ferramenta que podem ser usados para o dispositivo definido. Os parâmetros a seguir são mostrados: o **ID** (número de identificação exclusivo) da ferramenta, o **nome do atalho** que é usado para fazer referência ao atalho na biblioteca de objetos e o **nome do arquivo executável** da ferramenta. O botão **Avançado** expande a caixa de diálogo e fecha a extensão da mesma:

A caixa de diálogo expandida mostra as propriedades globais do atalho conforme definido no arquivo do dispositivo. Mais tarde, um campo de edição fica disponível sempre que um **diretório** de trabalho pode ser definido, o qual deve ser usado para ações do arquivo executável. O caminho será salvo sem fechar a caixa de diálogo assim que o botão **Aplicar** for pressionado.

Propriedades da ferramenta:

FixedCount	Número de atalhos da ferramenta, que são inseridos de forma inalterada e automaticamente na pasta Tools. Somente se "0" é inserido aqui, o usuário poderá criar quantos atalhos desejar. Por favor considere: para atalhos que são definidos como "fixos" pelo arquivo do dispositivo, o número de uso possível na pasta Ferramentas é predeterminado e as propriedades não podem ser modificadas pelo usuário MasterTool IEC ("<R>" no Organizador de Objetos).
EXE-Nome:	Nome do arquivo ou caminho completo do arquivo executável da ferramenta. Aqui também pode-se inserir um caminho do Registro apontando para um arquivo EXE: "[caminho do Registro]. < entrada do registro neste caminho apontando para um arquivo EXE>" Se não houver nenhuma entrada, a extensão de arquivo, que é fornecida no "Modelo do Parâmetro," automaticamente iniciará, via Windows, o arquivo EXE da ferramenta correspondente. Exemplos: "C:\programa\notepad.exe", "345.PDF"
DefaultDisplayName :	Nome que é usado para representar as ferramentas do Organizador de Objetos. Possivelmente, o modelo $ (número da instância) é usado (veja abaixo 'Modelo do Parâmetro').
Modelo de parâmetro:	Modelos para determinar o arquivo que deve ser aberto pela ferramenta. Os modelos a seguir podem ser usados via caracteres especiais apropriados: $(PROJECT_NAME) Nome do projeto aberto no momento (nome do arquivo sem extensão * .PRO ") $(PROJECT_PATH) Caminho do diretório onde o arquivo de projeto está (sem indicação da unidade) $(PROJECT_DRIVE) Unidade onde o projeto aberto no momento está

	$(COMPILE_DIR) Diretório de compilação do projeto (incluindo indicação da unidade) $(TOOL_EXE_NAME) Nome do arquivo EXE da ferramenta $(DISPLAY_NAME) Nome do atalho atual, como definido na pasta 'Ferramentas' $(INSTANCE_NUMBER) Número do atalho (número da instância, número de execução, começando com "1") $(MASTERTOOLIEC_EXE_DIR) Caminho do diretório onde o arquivo MasterTool IEC EXE está (incluindo a identificação do drive) A conversão de um modelo é obtida na caixa de diálogo para as propriedades de atalho (ver abaixo) Exemplo: "$ (PROJECT_NAME) _ $ (INSTANCE_NUMBER) .cfg" O arquivo CFG com o < nome do projeto MasterTool IEC atual > _ < número de atalho >.cfg será aberto na ferramenta.
Baixar modelos de arquivo:	Arquivos, caminhos de arquivos e modelos para arquivos que serão copiados para o CP durante o download. Se a opção editável estiver ativada, a lista desses arquivos poderá ser editada na caixa de diálogo de propriedades do atalho. Se um nome de arquivo é inserido sem caminho, o arquivo será pesquisado na pasta onde está o arquivo EXE MasterTool IEC. Exemplo: "a.up;$(PROJECT_NAME).zaw;$(INSTANCE_NUMBER).upp" Os arquivos a.up, <projeto MasterTool IEC atual>.pro e <número do atalho>.upp serão copiados para o CP no próximo download

Tabela 6-5. Propriedades da ferramenta

2. Propriedades do Atalho

Marque uma entrada de atalho na Árvore de 'Ferramentas' no Organizador de Objetos e selecione o comando 'Propriedades do Objeto' no menu de contexto ou no menu 'Projeto' 'Objeto'. A caixa de diálogo 'Propriedades do Atalho' será aberta, contendo os seguintes itens:

Comando	Chamando a ferramenta... Caminhos do arquivo EXE e do arquivo que é nomeado em 'Parâmetros' (predefinida pelo 'Modelo do Parâmetro', veja acima). Por exemplo: D:\listings\textfile.txt C:\programs\notepad.exe
Parâmetro	Caminho do arquivo que deve ser chamado pela ferramenta. Este é definido no arquivo do dispositivo e pode ser editado aqui, se a opção 'editável' (veja abaixo) tiver sido ativada.
Arquivos para download	Primeiro encontrar-se-ão aqui os nomes de arquivos que são definidos pelo arquivo do dispositivo e que também são descritos nas propriedades da ferramenta (veja acima). Se a opção 'editável' está ativada na caixa de diálogo estendida (veja abaixo), pode-se modificar a lista. Para essa finalidade pressione botão Novo para abrir a caixa de diálogo 'Nome do Arquivo', onde pode-se inserir outro arquivo e um caminho de arquivo. Ao inserir um arquivo sem caminho, ele será pesquisado no diretório onde está o arquivo EXE MasterTool IEC. O botão de exclusão removerá a entrada da lista selecionada.

Tabela 6-6. Propriedades do Atalho

O botão **padrão** redefine as entradas da caixa de diálogo com os valores padrão definidos pelo arquivo do dispositivo.

O botão **Aplicar** salva as configurações feitas sem fechar a caixa de diálogo das propriedades.

O botão **Avançado** expande a caixa de diálogo da seguinte maneira:

Diálogo 'Propriedades avançadas' de um atalho:

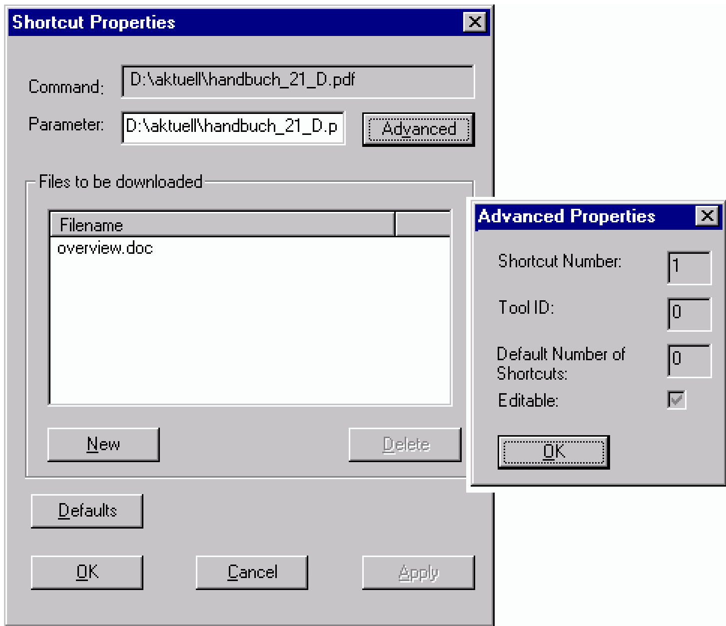


Figura 6-30. Diálogo 'Propriedades avançadas' de um atalho

Número do Atalho	Número em execução, começando com 1. Novos atalhos da ferramenta atual assumirão o próximo número mais alto. Se um atalho for removido posteriormente, os números dos atalhos restantes ficarão inalterados. O número de atalho pode ser inserido em outras definições usando o modelo $(INSTANCE_NUMBER) (por exemplo, consultar acima, 'Modelos do Parâmetro').
ID da Ferramenta	Número de identificação exclusiva da ferramenta; definido no arquivo do dispositivo.
Número Padrão de Atalhos	Número de atalhos (Instâncias) para uma ferramenta. Corresponde ao "FixedCount" definido no arquivo do dispositivo. Consultar a ferramenta Propriedades acima.
Editável	Se esta opção estiver ativada, será possível editar o campo 'Parâmetros' e a lista de arquivos que devem ser baixadas.

Tabela 6-7. Diálogo 'Propriedades avançadas' de um atalho

O botão **OK** aplica as configurações feitas e fecha a caixa de diálogo das propriedades.

Gerenciando atalhos de ferramenta

Criando novos atalhos de ferramenta

Selecione a entrada “Ferramentas” ou uma entrada do atalho na árvore dos recursos do Organizador de objetos e dê o comando “Acrescentar objeto” no menu do contexto ou no menu “Projeto” “Objeto” para abrir a caixa de diálogo “**Criar atalho**”:

A tabela lista todas as ferramentas para as quais os atalhos novos (conexões) podem ser criados. De acordo com as definições no arquivo do dispositivo, os seguintes parâmetros são indicados: **ID** da ferramenta, **Nome padrão do atalho** e o **Nome do arquivo executável**.

A fim de criar um/outro atalho para uma das ferramentas oferecidas, selecione esta ferramenta via clique no mouse na coluna “ID”. Então pode-se modificar o nome de padrão do atalho no campo de edição Nome do atalho e confirmá-lo com OK. Isto funcionará somente se for definido um nome que não foi usado ainda.

Caixa de diálogo “Criar atalho”:

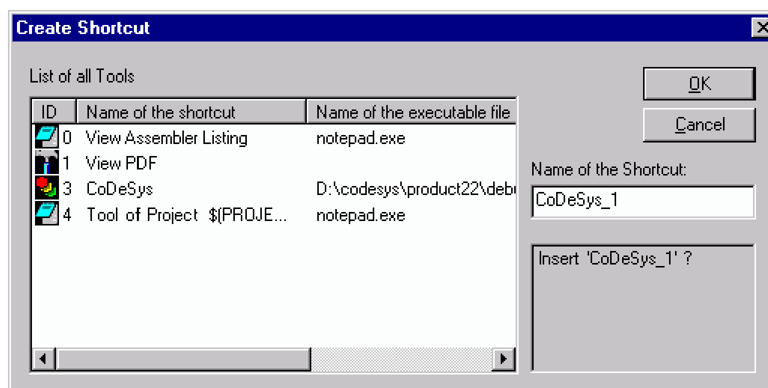


Figura 6-31. Caixa de diálogo "Criar atalho"

OK fecha a caixa de diálogo e o atalho novo será introduzido na árvore dos recursos, representada pelo nome do atalho e por um número do atalho que seja 1 mais elevado do que o mais elevado usado para um modelo desta ferramenta até agora.

Na área abaixo do campo do nome, dicas referentes a entradas de usuários serão exibidas.

Excluindo atalhos da ferramenta

O atalho pode ser excluído por meio do comando de exclusão no menu de contexto (botão direito do mouse) ou no menu 'Projeto' 'Objeto'. O comando só está disponível, se não houver nenhum número fixo de atalhos definido para o atalho selecionado. Os números de atalho dos atalhos restantes não serão alterados ao excluir um atalho.

Executando atalhos de ferramenta

Um atalho será executado com um clique duplo na entrada da árvore de recursos ou via comando 'Abrir objeto' no menu 'Projeto' 'Objeto' no menu de contexto (botão direito do mouse).

Se a execução do arquivo, que é definido nas propriedades do atalho (Parâmetro) falhar, então, uma mensagem de erro apropriada será exibida. Se um arquivo de parâmetro não for encontrado, o arquivo EXE da ferramenta será executado e uma caixa de diálogo será aberta, perguntando se o arquivo deve ser criado.

Se o arquivo EXE da ferramenta não for encontrado no caminho definido ou se nenhum caminho tiver sido definido, a caixa de diálogo padrão para selecionar um arquivo será aberta e será solicitado ao usuário que insira o caminho do arquivo EXE. Esse caminho será salvo quando a caixa de diálogo é fechada pelo OK e, portanto, estará disponível para a ferramenta também em outros projetos MasterTool IEC.

Salvando Atalhos da Ferramenta

Quando o projeto MasterTool IEC é salvo, o status e a configurações da pasta 'Ferramentas' na árvore de recursos também são salvos.

Nota: ao salvar um projeto por 'Salvar como' com um novo nome, deve-se considerar o que segue. Ao usar o modelo $(PROJECT_NAME) na definição dos arquivos de parâmetros e dos arquivos para download, se foram adicionados atalhos para uma ferramenta (FixedCount = 0) no projeto antigo, no novo projeto os nomes de arquivo precisam ser renomeados manualmente correspondendo ao novo nome do projeto. Em contrapartida, para uma ferramenta que é definida com um número fixo de atalhos, o modelo sempre será substituído automaticamente pelo nome do projeto atual!

Perguntas freqüentes em ferramentas

Por que não obtenho nenhuma entrada 'Ferramentas' nos 'Recursos'?

A funcionalidade 'Ferramentas' estará disponível somente se estiver definida no arquivo do dispositivo do sistema do dispositivo em uso.

Para quais ferramentas os atalhos já estão disponíveis e quais atalhos posso adicionar?

Abra a pasta 'Ferramentas' na árvore de 'Recursos' do Organizador de Objetos com um clique duplo no sinal de mais. Aparecerá quais ferramentas já estão conectadas ao projeto atual. Se foi configurado um novo projeto e ainda não foi trabalhado na lista de ferramentas, então, apenas aquelas entradas serão exibidas, que são predefinidas sem alteração pelas definições no arquivo do dispositivo. Caso contrário, se poderá ver uma lista de ferramentas já modificada especificamente para o projeto. Para verificar se a lista é extensível para novas entradas, selecione o comando 'Acrescentar objeto'. Obtém-se uma caixa de diálogo oferecendo todas as ferramentas para as quais atalhos adicionais podem ser criados.

Quais propriedades globais têm as ferramentas disponíveis?

Selecione a entrada 'Ferramentas' no Organizador de Objetos e escolha o comando 'Propriedades do objeto' no menu de contexto (botão direito do mouse). Expanda a caixa de diálogo pressionando o botão 'Avançado'. Neste momento aparece uma lista das ferramentas disponíveis e os parâmetros correspondentes. Selecione uma das ferramentas com um clique no mouse no símbolo ID para – por exemplo - obter o número permitido de atalhos para a ferramenta no campo apropriado ou para obter quais arquivos serão baixados para o CP se o atalho está ativado. Os nomes de arquivo ou caminhos podem ser mostrados na forma de Modelos, que serão interpretados para cada atalho único conforme descrito no seguinte parágrafo:

Quais propriedades individuais têm os atalhos disponíveis?

Selecione uma das entradas abaixo de 'Ferramentas' no Organizador de Objetos e selecione o comando 'Propriedades do Objeto' no menu de contexto (botão direito do mouse). Pressione o botão 'Avançado' para obter os parâmetros do atalho escolhido. Parcialmente eles corresponderão às propriedades globais das ferramentas acima descritas. Se permitido pela definição no arquivo do dispositivo, pode-se editar esses parâmetros aqui.

Como posso criar um atalho para uma ferramenta?

Selecione a entrada 'Ferramentas' no Organizador de Objetos e escolha o comando 'Acrescentar objeto' no menu de contexto (botão direito do mouse). Aparecerá uma lista das ferramentas disponíveis (aquelas para as quais o número máximo de atalhos ainda não foi atingido). Escolha uma ferramenta e pressione ' OK '. A ferramenta agora será inserida na pasta Ferramentas no Organizador de Objetos. Para inseri-la mais uma vez, então, insira um nome de ferramenta diferente em primeiro lugar, o que significa marcar a nova entrada como outra instância da mesma ferramenta. Por exemplo, pode-se nomear as instâncias da ferramenta Toolxy como "Toolxy_1", " Toolxy_2 ", etc.

Como eu posso modificar os parâmetros de uma ferramenta?

Para modificar os parâmetros de um atalho (instância de uma conexão de ferramenta), selecione o atalho no Organizador de Objetos e escolha o comando 'Propriedades do objeto' no menu de contexto. Depende da pré-definição da ferramenta no arquivo do dispositivo, se os parâmetros podem ser editados na caixa de diálogo de propriedades. (Consultar a caixa de diálogo expandida se a opção específica está ativada. O botão 'Default' zera todos os valores editados para os padrões.

Como eu posso executar um atalho de ferramenta?

Faça um duplo clique na entrada do atalho no Organizador de Objetos ou selecione o comando 'Abrir objeto' no menu de contexto ou no menu 'Projeto' quando a entrada estiver selecionada no Organizador de Objetos.

7. Apêndices

Apêndice A: operadores IEC e funções adicionais

Operadores aritméticos

ADD

Adição de variáveis dos tipos: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL e LREAL.

Duas variáveis de TEMPO podem também ser adicionadas em outro tempo. (por exemplo, t#45s + t#50s = t#1m35s)

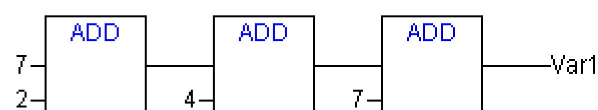
Exemplo em IL:

```
LD  7
ADD 2,4,7
ST  VAR1
```

Exemplo em ST:

```
VAR1 := 7+2+4+7;
```

Exemplo em FBD:



MUL

Multiplicação das variáveis dos tipos: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL e LREAL.

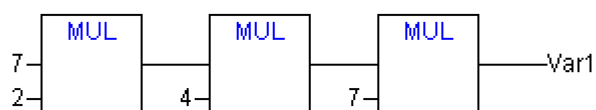
Exemplo em IL:

```
LD  7
MUL 2,4,7
ST  VAR1
```

Exemplo em ST:

```
VAR1 := 7*2*4*7;
```

Exemplo em FBD:



SUB

Subtração de uma variável (por outra), dos tipos: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL e LREAL.

Uma variável de tempo (TIME) também pode ser subtraída de outra variável de tempo, resultando em terceira variável de tipo tempo. Note que os valores negativos de tempo são indefinidos.

Exemplo em IL:

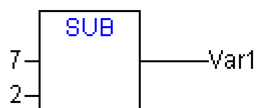
```
LD  7
SUB 2
```

ST VAR1

Exemplo em ST:

VAR1 := 7-2;

Exemplo em FBD:



DIV

Divisão de uma variável por outra, dos tipos: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL e LREAL.

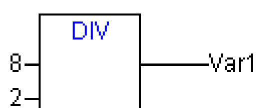
Exemplo em IL:

```
LD 8
DIV 2
ST VAR1 (*RESULTADO É 4 *)
```

Exemplo em ST:

VAR1 := 8/2;

Exemplo em FBD:



Nota: ao definir funções no projeto com os nomes CheckDivByte, CheckDivWord, CheckDivDWord e CheckDivReal, pode-se utilizá-las para verificar o valor do divisor, ao usar o operador DIV, por exemplo, para evitar uma divisão por 0. As funções devem ter os nomes listados acima.

Atenção: considerar que sistemas do dispositivo diferentes podem comportar-se de forma diferente em uma divisão por zero!

Consultar a seguir um exemplo para a execução da função CheckDivReal:

Exemplo para a implementação da função CheckDivReal:

```
FUNCTION CHECKDIVREAL : REAL
VAR_INPUT
 DIVISOR:REAL;
END_VAR
IF DIVISOR = 0 THEN
 CHECKDIVREAL:=1;
ELSE
 CHECKDIVREAL:=DIVISOR;
END_IF;
```

O operador DIV usa a saída de função CheckDivReal como divisor. Em um programa como mostrado no exemplo seguinte, isto evita uma divisão por 0, o divisor (d) é definido de 0 a 1. Assim, o resultado da divisão é 799.

```
PROGRAM PLC_PRG
VAR
 ERG:REAL;
 V1:REAL:=799;
 D:REAL;
END_VAR
ERG:= V1 / D;
```

Atenção: as funções CheckDiv fornecidas pela biblioteca Check.Lib, são apenas exemplos de soluções! Antes de usar esses módulos de biblioteca verifique se eles trabalham no seu contexto, ou execute as funções adequadas diretamente como uma POU no projeto.

MOD

Módulo da divisão de uma variável por outra, dos tipos: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT. O resultado desta função será o resto da divisão. Este resultado será um número inteiro.

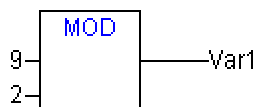
Exemplo em IL:

```
LD  9
MOD 2
ST  VAR1 (*RESULTADO É 1 *)
```

Exemplo em ST:

```
VAR1 := 9 MOD 2;
```

Exemplo em FBD:

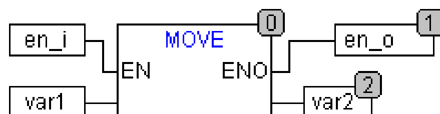


MOVE

Atribuição de uma variável à outra variável de um tipo adequado. Como MOVE está disponível como uma caixa nos editores gráficos LD, CFC, lá a funcionalidade EN/ENO também pode ser aplicada em uma atribuição de variável. No editor FBD, entretanto, isto não é possível.

Exemplo em CFC em conjunto com a função EN/ENO:

Somente se en_i for TRUE, var1 será atribuída à var2.



Exemplo em IL:

```
LD IVAR1
MOVE
ST IVAR2 (* RESULTADO: NA IVAR2 ÉH ATRIBUÍDO O VALOR DE IVAR1 *)
```

Obtém-se o mesmo resultado com:

```
LD IVAR1
ST IVAR2 )
```

Exemplo em ST:

```
IVAR2 := MOVE(IVAR1);
```

Obtém-se o mesmo resultado com: ivar2 := ivar1;.

INDEXOF

Esta função não foi prevista pela norma IEC61131-3.

Execute esta função para encontrar o índice interno para uma POU.

Exemplo em ST:

```
VAR1 := INDEXOF(POU2);
```

SIZEOF

Esta função não foi prevista pela norma IEC61131-3.

Execute esta função para determinar o número de bytes solicitados pela variável fornecida.

Exemplo em IL:

```
ARR1:ARRAY[0..4] OF INT;
VAR1 INT
LD ARR1
SIZEOF
ST VAR1 (*RESULTADO É 10 *)
```

Exemplo em ST:

```
VAR1 := SIZEOF(ARR1);
```

Operadores de cadeia de bits

AND

AND binário de operandos de bit. Os operandos devem ser do tipo BOOL, BYTE, WORD ou DWORD.

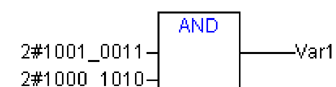
Exemplo em IL:

```
VAR1 BYTE
LD 2#1001_0011
AND 2#1000_1010
ST VAR1 (*RESULTADO É 2#1000_0010 *)
```

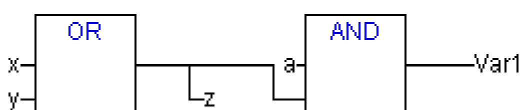
Exemplo em ST:

```
VAR1 := 2#1001_0011 AND 2#1000_1010
```

Exemplo em FBD:



Considerar: para um passo de programa no SFC, como segue...



Se forem usados geradores 68xxx ou de código-C, observe que a alocação do valor da segunda variável de entrada no módulo operador AND para a variável z não será executada! Isto é devido ao processamento otimizado no SFC, no caso de valor FALSE na variável de entrada.

OR

OR binário de operandos de bit. Os operandos devem ser do tipo BOOL, BYTE, WORD ou DWORD.

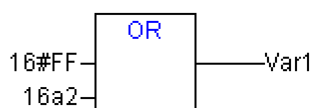
Exemplo em IL:

```
VAR1 :BYTE;
LD 2#1001_0011
OR 2#1000_1010
ST  VAR1 (*RESULTADO É 2#1001_1011 *)
```

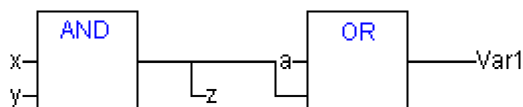
Exemplo em ST:

```
VAR1 := 2#1001_0011 OR 2#1000_1010
```

Exemplo em FBD:



Considerar: para um passo de programa no SFC, como segue...



Se forem usados geradores 68xxx ou de código-C, observe que a alocação do valor da segunda variável de entrada no módulo operador OR para a variável z não será executada! Isto é devido ao processamento otimizado no SFC, no caso de valor FALSE na variável de entrada.

XOR

XOR binário de operandos de bit. Os operandos devem ser do tipo BOOL, BYTE, WORD ou DWORD.

Obs.: considere o comportamento da função XOR na forma estendida, isso significa que, se houver mais de 2 entradas, as entradas serão verificadas em pares e os resultados específicos serão comparados novamente em pares (isto cumpre a norma, mas pode não ser esperado pelo usuário).

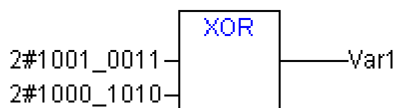
Exemplo em IL:

```
VAR1 :BYTE;
LD 2#1001_0011
XOR 2#1000_1010
ST  VAR1 (*RESULTADO É 2#0001_1001 *)
```

Exemplo em ST:

```
VAR1 := 2#1001_0011 XOR 2#1000_1010
```

Exemplo em FBD:



NOT

NOT binário de operandos de bit. Os operandos devem ser do tipo BOOL, BYTE, WORD ou DWORD.

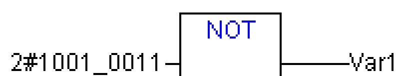
Exemplo em IL:

```
VAR1 :BYTE;
LD 2#1001_0011
NOT
ST  VAR1 (*RESULTADO É 2#0110_1100 *)
```

Exemplo em ST:

VAR1 := NOT 2#1001_0011

Exemplo em FBD:



Operadores de deslocamento de bits

SHL

Deslocamento de bits à esquerda de um operando: erg: = SHL (in, n) é deslocado à esquerda por n bits. Se n > largura do tipo de dados, para BYTE, WORD e DWORD será preenchida com zeros. Mas se tipos de dados com sinal são usados (por exemplo: INT), então, um deslocamento aritmético será executado em tais casos, isso significa que o operando será preenchido com o valor do bit mais alto.

Nota: por favor, observe que o montante de bits, que é considerado para a operação aritmética, é definido pelo tipo de dados da variável de entrada! Se a variável de entrada é uma constante, o menor tipo de dados possível é considerado. O tipo de dados da variável de saída não tem efeito sobre a operação aritmética.

Nota: veja no exemplo a seguir em notação hexadecimal que obtém-se resultados diferentes para erg_byte e erg_word, dependendo do tipo de dados da variável de entrada (BYTE ou WORD), embora os valores das variáveis de entrada in_byte e in_word sejam os mesmos.

Exemplo em ST:

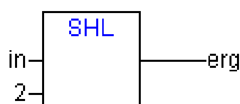
```
PROGRAM SHL_ST

VAR
IN_BYTE : BYTE:=16#45;
IN_WORD : WORD:=16#45;
ERG_BYTE : BYTE;
ERG_WORD : WORD;
N: BYTE :=2;
END_VAR

ERG_BYTE:=SHL(IN_BYTE,N); (*RESULTADO É 16#14 *)

ERG_WORD:=SHL(IN_WORD,N); (*RESULTADO É  16#01141 *)
```

Exemplo em FBD:



Exemplo em IL:

```
 LD  16#45

SHL 2

ST  ERG_BYTE
```

SHR

Deslocamento de bits à esquerda de um operando: erg: = SHR (in, n) é deslocado para a direita por n bits. Se n > largura do tipo de dados, para BYTE, WORD e DWORD será preenchido com zeros. Mas se tipos de dados com sinal são usados, como, por exemplo, INT, então, um deslocamento aritmético será executada em tais casos, isso significa que o operando será preenchido com o valor do bit mais alto.

Nota: por favor, note que o montante de bits, que é considerado para a operação aritmética, é definido pelo tipo de dados da variável de entrada! Se a variável de entrada é uma constante, o menor tipo de dados possível é considerado. O tipo de dados da variável de saída não tem efeito sobre a operação aritmética.

Nota: veja no exemplo a seguir em notação hexadecimal os resultados da operação aritmética, dependendo do tipo da variável de entrada (BYTE ou WORD).

Exemplo em ST:

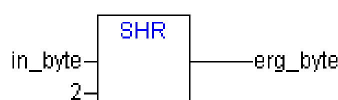
```
PROGRAM SHR_ST

VAR
IN_BYTE : BYTE:=16#45;
IN_WORD : WORD:=16#45;
ERG_BYTE : BYTE;
ERG_WORD : WORD;
N: BYTE :=2;
END_VAR

ERG_BYTE:=SHR(IN_BYTE,N); (* RESULTADO É 11 *)

 ERG_WORD:=SHR(IN_WORD,N); (* RESULTADO É 0011 *)
```

Exemplo em FBD:



Exemplo em IL:

```
LD  16#45

SHR 2

ST  ERG_BYTE
```

ROL

Rotação de bits de um operando a esquerda: erg: = ROL (in, n) erg. in e n devem se do tipo BYTE, WORD e DWORD . ‘in’ será deslocada em uma posição de bit à esquerda n vezes, enquanto o bit que está mais a esquerda será reinserido a partir da direita.

Nota: observe que o montante de bits, que é considerado para a operação aritmética, é definido pelo tipo de dados da variável de entrada! Se a variável de entrada é uma constante, o menor tipo de dados possível é considerado. O tipo de dados da variável de saída não tem nenhum efeito sobre a operação aritmética.

Veja no exemplo a seguir em notação hexadecimal que obtém-se resultados diferentes para erg_byte e erg_word, dependendo do tipo de dados da variável de entrada (BYTE ou WORD), embora os valores das variáveis de entrada in_byte e in_word sejam os mesmos.

Exemplo em ST:

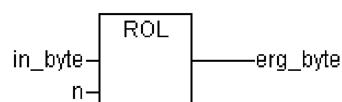
```
PROGRAM ROL_ST

VAR
IN_BYTE : BYTE:=16#45;
IN_WORD : WORD:=16#45;
ERG_BYTE : BYTE;
ERG_WORD : WORD;
N: BYTE :=2;
END_VAR

ERG_BYTE:=ROL(IN_BYTE,N); (*RESULTADO É 16#15 *)

ERG_WORD:=ROL(IN_WORD,N); (*RESULTADO É 16#0114 *)
```

Exemplo em FBD:



Exemplo em IL:

```
LD  16#45

ROL 2

ST  ERG_BYTE
```

ROR

Rotação de bits de um operando à direita: erg: = ROL (in, n) erg, in e n devem ser do tipo BYTE, WORD e DWORD. ‘in’ será deslocada uma posição de bit à direita n vezes, enquanto o bit que está mais a esquerda será reinserido a partir da esquerda.

> Nota: por favor, observe que o montante de bits, que é considerado para a operação aritmética, é definido pelo tipo de dados da variável de entrada! Se a variável de entrada é uma constante, o menor tipo de dados possível é considerado. O tipo de dados da variável de saída não tem efeito sobre a operação aritmética.

Veja no exemplo a seguir em notação hexadecimal que obtém-se resultados diferentes para erg_byte e erg_word, dependendo do tipo de dados da variável de entrada (BYTE ou WORD), embora os valores das variáveis de entrada in_byte e in_word sejam os mesmos.

Exemplo em ST:

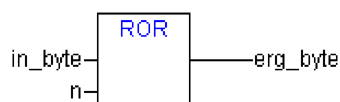
```
PROGRAM ROR_ST

VAR
IN_BYTE : BYTE:=16#45;
IN_WORD : WORD:=16#45;
ERG_BYTE : BYTE;
ERG_WORD : WORD;
N: BYTE :=2;
END_VAR

ERG_BYTE:=ROR(IN_BYTE,N); (*RESULTADO É 16#51 *)

ERG_WORD:=ROR(IN_WORD,N); (*RESULTADO É 16#4011 *)
```

Exemplo em FBD:



Exemplo em IL:

```
LD 16#45

ROR 2

ST ERG_BYTE
```

Operadores de seleção

Todas as operações de seleção também podem ser executadas com variáveis. Para maior clareza, nós limitaremos nossos exemplos as que utilizam constantes como operadores.

SEL

Seleção binária.

OUT := SEL(G, IN0, IN1) significa:

OUT := IN0 if G=FALSE;

OUT := IN1 if G=TRUE.

IN0, IN1 e OUT podem ser quaisquer tipos de variável, G deve ser BOOL. O resultado da seleção é IN0 se G is FALSE, IN1 se G é TRUE.

Exemplo em IL:

```
LD  TRUE

SEL 3,4   (* IN0 = 3, IN1 =4 *)

ST  VAR1  (*RESULTADO É 4 *)

LD  FALSE

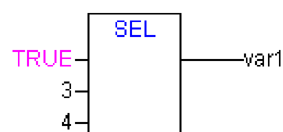
SEL 3,4

ST  VAR1  (*RESULTADO É  3 *)
```

Exemplo em ST:

```
VAR1:=SEL (TRUE,3,4); (*RESULTADO É  4 *)
```

Exemplo em FBD:



Nota: observe que uma expressão anterior a IN1 ou IN2 não será processada se IN0 for TRUE.

MAX

Função máximo. Retorna o maior dos dois valores.

OUT := MAX(IN0, IN1)

IN0, IN1 e OUT podem ser quaisquer tipos de variável.

Exemplo em IL:

LD 90

MAX 30

MAY 40

MAX 77

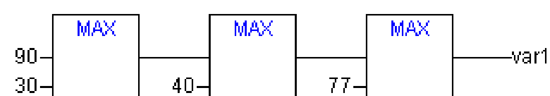
ST VAR1 (*RESULTADO É 90 *)

Exemplo em ST:

VAR1:=MAX(30,40); (*RESULTADO É 40 *)

VAR1:=MAX(40,MAX(90,30)); (*RESULTADO É 90 *)

Exemplo em FBD:



MIN

Função mínimo. Retorna o menor dos dois valores.

OUT := MIN(IN0, IN1)

IN0, IN1 e OUT podem ser quaisquer tipos de variável.

Exemplo em IL:

LD 90

MIN 30

MIN 40

MIN 77

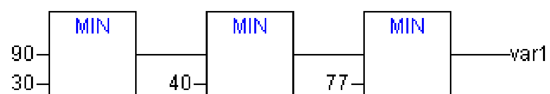
ST VAR1 (*RESULTADO É 30 *)

Exemplo em ST:

VAR1:=MIN (90,30); (*RESULTADO É 30 *);

VAR1:=MIN (MIN(90,30),40); (*RESULTADO É 30 *);

Exemplo em FBD:



LIMIT

“Limite”

OUT: = LIMIT(Min, IN, Max) means:

OUT: = MIN (MAX (IN, Min), Max)

Max é o limite superior e Min é o limite inferior para o resultado. Se o valor IN exceder o limite superior máximo, LIMIT irá retornar Max. Se IN for inferior ao mínimo, o resultado será Min.

IN e OUT podem ser quaisquer tipos de variáveis.

Exemplo em IL:

```
LD 90

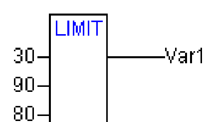
LIMIT 30,80

ST VAR1 (*RESULTADO É 80 *)
```

Exemplo em ST:

```
VAR1:=LIMIT (30,90,80); (* RESULTADO E 80 *)
```

Exemplo em FBD:



MUX

Multiplexador.

OUT: = MUX(K, IN0,...,INn) significa:

OUT:= INK.

IN0, ...INn e OUT podem ser qualquer tipo de variável. K deve ser BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT ou UDINT. MUX seleciona o K-ésimo valor entre um grupo de valores.

Exemplo em IL:

```
LD 0

MUX 30, 40,50,60,70,80

ST VAR1 (*RESULTADO É 30 *)
```

Exemplo in ST:

```
VAR1:=MUX (0 ,30,40,50,60,70,80); (*RESULTADO É 30 *);
```

Nota: observe que uma expressão que ocorre à frente de outra entrada que não INK, não será processada para otimizar o tempo de execução! Somente no modo de simulação serão executadas todas as expressões.

Operadores de comparação

GT

“Maior que”

Um operador booleano retorna o valor TRUE quando o valor do primeiro operando é superior ao do segundo. Os operandos podem ser BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME e STRING.

Exemplo em IL:

LD 20

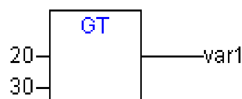
GT 30

ST VAR1 (*RESULTADO É FALSE *)

Exemplo em ST:

VAR1:= 20 > 30 > 40 > 50 > 60 > 70;

Exemplo em FBD:



LT

"Menor que"

Um operador booleano retorna o valor TRUE quando o valor do primeiro operando é menor que o do segundo. Os operandos podem ser BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME e STRING.

Exemplo em IL:

LD 20

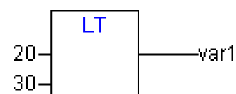
LT 30

ST VAR1 (*RESULTADO É TRUE *)

Exemplo em ST:

VAR1:= 20 < 30;

Exemplo em FBD:



LE

"Menor ou igual a"

Um operador booleano retorna o valor TRUE quando o valor do primeiro operando é menor ou igual ao do segundo. Os operandos podem ser BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME e STRING.

Exemplo em IL:

LD 20

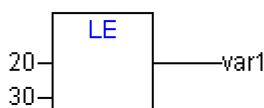
LE 30

ST VAR1 (* RESULTADO É TRUE *)

Exemplo em ST:

VAR1:= 20 <= 30;

Exemplo em FBD:



GE

"Maior ou igual a"

Um operador booleano retorna o valor TRUE quando o valor do primeiro operando é maior ou igual ao do segundo. Os operandos podem ser BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME e STRING.

Exemplo em IL:

```
LD 60

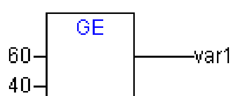
GE 40

ST VAR1 (* RESULTADO É TRUE *)
```

Exemplo em ST:

```
VAR1:= 60 >= 40;
```

Exemplo em FBD:



EQ

"Igual a"...

Um operador booleano retorna o valor TRUE quando o valor do operando é igual. Os operandos podem ser BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME e STRING.

Exemplo em IL:

```
LD 40

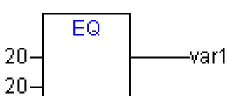
EQ 40

ST VAR1 (* RESULTADO É TRUE *)
```

Exemplo em ST:

```
VAR1:= 40 = 40;
```

Exemplo em FBD:



NE

"Diferente de"

Um operador booleano retorna o valor TRUE quando o valor do operando é diferente. Os operandos podem ser BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, DATE, TIME_OF_DAY, DATE_AND_TIME e STRING.

Exemplo em IL:

LD 40

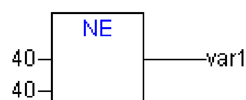
NE 40

ST VAR1 (* RESULTADO É FALSE *)

Exemplo em ST:

VAR1:= 40 <> 40;

Exemplo em FBD:



Operadores de endereço

Atenção: após uma alteração online pode haver alterações nos dados em determinados endereços. Por favor, neste caso, considere o uso de ponteiros em endereços.

ADR

Função endereço não prevista pela IEC61131-1.

ADR retorna o endereço dos seus argumentos em uma DWORD. Este endereço pode ser enviado para a criação de funções a serem tratadas como um ponteiro ou ele pode ser atribuído a um ponteiro dentro do projeto.

dwVar:=ADR(bVAR);

Exemplo em IL:

LD BVAR

ADR

ST DWVAR

MAN_FUN1

ADRINST

Função endereço não prevista pela IEC61131-1.

ADRINST pode ser usado em uma instância do bloco funcional para retornar os endereços desta instância em uma DWORD. Este endereço pode ser enviado para criação de funções a serem tratadas como um ponteiro ou ele pode ser atribuído a um ponteiro dentro do projeto.

Exemplos em ST (dentro de uma instância do bloco funcional):

DVAR:=ADRINST (); (* ENDEREÇO DA INSTÂNCIA É ESCRITO NA VARIÁVEL DVAR *)

FUN (A:=ADRINST()); (* O ENDEREÇO DA INSTÂNCIA É FORNECIDO PARA UM PARÂMETRO DE ENTRADA DE UMA FUNÇÃO FUN *)

Exemplos em IL:

ADRINST

ST DVAR

ADRINST

FUN

BITADR

Função endereço não prevista pela IEC61131-1.

BITADR retorna o offset do bit dentro do segmento em uma DWORD. Considere que o valor do offset depende se a opção endereçamento de byte nas configurações do dispositivo está ativada ou não.

```
VAR
VAR1 AT %IX2.3:BOOL;
BITOFFSET: DWORD;

END_VAR
```

Exemplo em IL:

```
BITOFFSET:=BITADR (VAR1); (* RESULTADO SE ENDEREÇO DE BYTE FOR=TRUE: 19, SE ENDEREÇO
DE BYTE FOR=FALSE: 35 *)
```

Exemplo em ST:

```
LD VAR1

BITADR

ST VAR2
```

Operadores de conteúdo

Um ponteiro pode ser de-referenciado adicionando " ^ " (operador de conteúdo), após o identificador de ponteiro.

Exemplo em ST:

```
PT: POINTER TO INT;

VAR_INT1: INT;

VAR_INT2: INT;

PT: = ADR(VAR_INT1);

VAR_INT2:=PT^;
```

Operadores de chamada

CAL

Chamada de um bloco funcional ou um programa

Use CAL em IL para chamar uma instância do bloco funcional. As variáveis que servirão como variáveis de entrada são colocadas entre parênteses logo após o nome da instância do bloco funcional.

Exemplo:

Chamada da instância Inst a partir de um bloco funcional, onde variáveis de entrada Par1 e Par2 são 0 e TRUE, respectivamente.

```
CAL INST(PAR1 := 0, PAR2 := TRUE)
```

Conversões de tipo

É proibida a conversão de um tipo de "maior" em um tipo "menor" (por exemplo, de INT para BYTE ou de DINT para WORD). Para fazer isso, é necessário solicitar um tipo especial de conversões. Pode-se basicamente converter qualquer tipo elementar em qualquer outro tipo elementar.

Sintaxe:

<elem.Typ1>_TO_<elem.Typ2>

Por favor, considerar que nas conversões TO_STRING a seqüência de caracteres é gerada justificada à esquerda. Se a STRING for definida para curta, será cortada do lado direito.

Conversões BOOL_TO

Conversão do tipo BOOL para qualquer outro tipo:

Para tipos de números o resultado é 1, quando o operando é TRUE e 0, quando o operando é FALSE.

Para o tipo STRING, o resultado é ‚TRUE' ou ‚FALSE'.

Exemplos em IL:

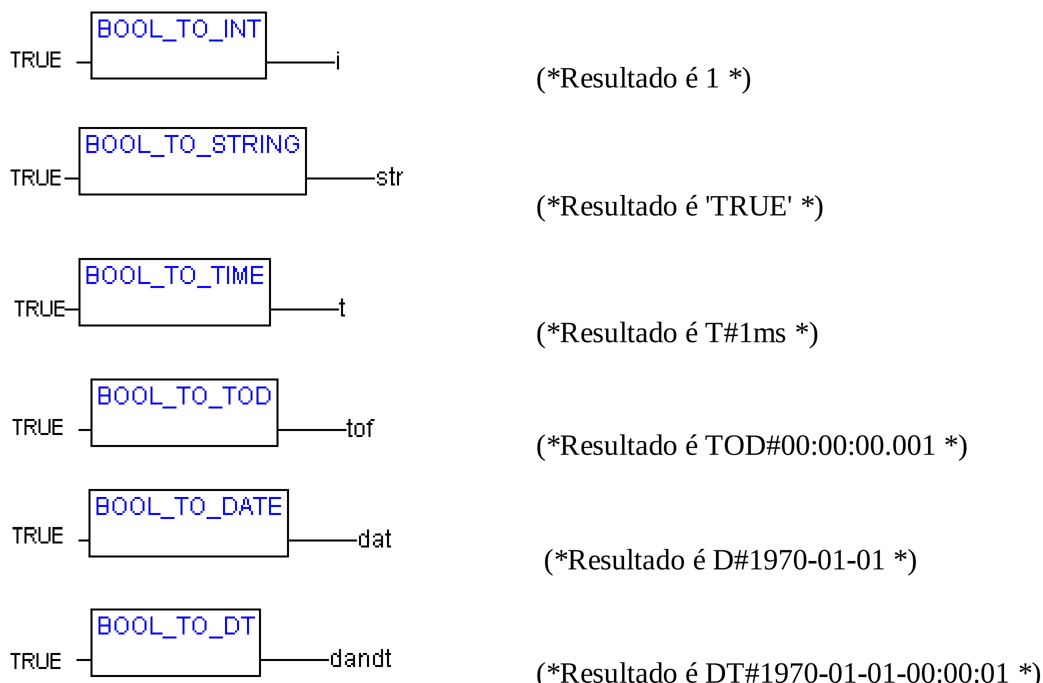
```
LD TRUE
BOOL_TO_INT
ST I (*RESULTADO É 1 *)
LD TRUE
BOOL_TO_STRING
ST STR      (*RESULTADO É 'TRUE' *)
LD TRUE
BOOL_TO_TIME
ST T(*RESULTADO É T#1MS *)
LD TRUE
BOOL_TO_TOD
ST   (*RESULTADO É TOD#00:00:00.001 *)
LD FALSE
BOOL_TO_DATE
ST DAT      (*RESULTADO É D#1970-01-01 *)
LD TRUE
BOOL_TO_DT
ST DANDT    (*RESULTADO É DT#1970-01-01-00:00:01 *)
```

Exemplos em ST:

```
I:=BOOL_TO_INT(TRUE);       (*RESULTADO É 1 *)
STR:=BOOL_TO_STRING(TRUE);      (*RESULTADO É "TRUE" *)
T:=BOOL_TO_TIME(TRUE);    (*RESULTADO É T#1MS *)
```

```
TOF:=BOOL_TO_TOD(TRUE);(*RESULTADO É TOD#00:00:00.001 *)
DAT:=BOOL_TO_DATE(FALSE);        (*RESULTADO É D#1970 *)
DANDT:=BOOL_TO_DT(TRUE);         (* RESULTADO É DT#1970-01-01-00:00:01 *)
```

Exemplos em FBD:



Conversões TO_BOOL

Conversão de outro tipo de variável para BOOL:

O resultado é TRUE quando o operando não for igual a 0. O resultado é FALSE quando o operando é igual a 0.

O resultado é TRUE para variáveis de tipo STRING, quando o operando é "TRUE", caso contrário o resultado é “FALSE”.

Exemplos em IL:

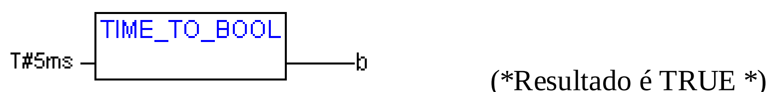
```
LD 213
BYTE_TO_BOOL
ST B          (*RESULT IS TRUE *)
LD 0
INT_TO_BOOL
ST B          (*RESULT IS FALSE *)
LD T#5MS
TIME_TO_BOOL
ST B          (*RESULT IS TRUE *)
LD 'TRUE'
STRING_TO_BOOL
```

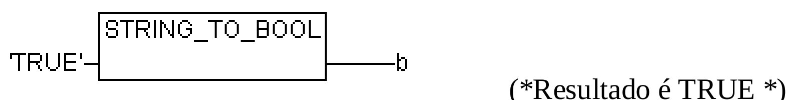
ST B (*RESULTADO É TRUE *)

Exemplos em FBD:









Exemplos em ST:

```
B = BYTE_TO_BOOL (2#11010101);     (*RESULTADO É TRUE *)

B: = INT_TO_BOOL (0);          (*RESULTADO É FALSE *)

B: = TIME_TO_BOOL ( T#5MS);          (*RESULTADO É TRUE *)

B: = STRING_TO_BOOL ('TRUE');       (*RESULTADO É TRUE *)
```

Conversão entre tipos de números inteiros

Conversão de um tipo de número inteiro para outro tipo de número...

Ao executar uma conversão de tipo de um maior para um menor, arrisca-se perder algumas informações. Se o número que está sendo convertido exceder o limite do intervalo, os primeiros bytes para o número serão ignorados.

Exemplo em ST:

```
SI := INT_TO_SINT(4223);
```

Ao salvar o inteiro 4223 (16# 107f representado em hex.) como uma variável SINT, ele aparecerá como 127 (16#7f representado em hex.).

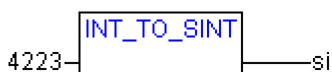
Exemplo em IL:

```
LD 2

INT_TO_REAL

MUL
```

Exemplo em FBD:



Conversões REAL_TO-/ LREAL_TO

Conversão de variável tipo REAL ou LREAL para um tipo diferente:

O valor será arredondado para cima ou para baixo para o número inteiro mais próximo e convertido no novo tipo de variável. Exceções a isto são os tipos de variáveis BOOL, STRING, REAL e LREAL.

Considerar na conversão para o tipo STRING, que o número total de dígitos é limitado a 16. Se o número (L)REAL contém mais dígitos, então, o décimo sexto será arredondado. Se o comprimento da STRING é definido para curto, ele será cortado iniciando a partir do final direito.

Ao executar uma conversão de tipo de um maior para um tipo menor, arrisca-se perder algumas informações.

Exemplo em ST:

```
I := REAL_TO_INT(1.5); (* RESULTADO É 2 *)

J := REAL_TO_INT(1.4); (*RESULTADO É 1 *)

I := REAL_TO_INT(-1.5); (*RESULTADO É -2 *)

J := REAL_TO_INT(-1.4); (*RESULTADO É -1 *)
```

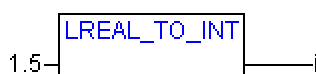
Exemplo em IL:

```
LD 2.7

REAL_TO_INT

GE %MW8
```

Exemplo em FBD:



Conversões TIME_TO/TIME_OF_DAY

Conversão da variável tipo TIME ou TIME_OF_DAY para um tipo diferente:

O tempo será armazenado internamente em uma DWORD em milissegundos (iniciando com 12: 00 para a variável TIME_OF_DAY). Este valor, então, será convertido.

Ao executar uma conversão de tipo de um maior para um tipo menor, arrisca-se perder algumas informações.

Para a variável do tipo STRING, o resultado é uma constante de tempo.

Exemplos em IL:

```
LD T#12MS

 TIME_TO_STRING

ST STR        (*RESULT IS 'T#12MS' *)

LD T#300000MS

TIME_TO_DWORD

ST DW         (*RESULT IS 300000 *)

LD TOD#00:00:00.012

TOD_TO_SINT

ST SI         (*RESULT IS 12 *)
```

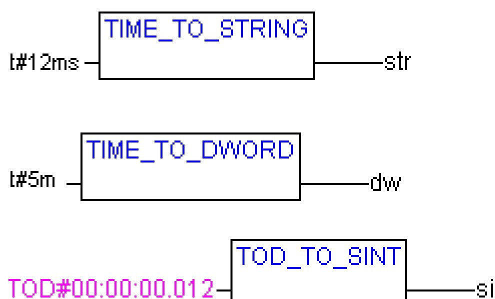
Exemplos em ST:

```
STR :=TIME_TO_STRING(T#12MS);     (* RESULT IS T#12MS *)
```

DW:=TIME_TO_DWORD(T#5M); (* RESULT IS 300000 *)

SI:=TOD_TO_SINT(TOD#00:00:00.012); (* RESULT IS 12 *)

Exemplos em FBD:



Conversões DATE_TO/DT_TO

Conversão da variável tipo DATA ou DATA_E_HORA para um tipo diferente:

A data será armazenada internamente em um DWORD em segundos desde 1 de Janeiro de 1970. Este valor, então, será convertido.

Ao executar uma conversão de um tipo maior para um tipo menor, arrisca-se perder algumas informações.

Para variáveis do tipo STRING, o resultado é a constante de data.

Exemplos em IL:

```
LD D#1970-01-01

DATE_TO_BOOL

ST B          (* RESULTADO É FALSE *)

LD D#1970-01-15

DATE_TO_INT

ST I (* RESULTADO É 29952 *)

LD DT#1970-01-15-05:05:05

DT_TO_BYTE

ST BYT        (*RESULTADO É 129 *)

LD DT#1998-02-13-14:20

DT_TO STRING

ST STR        (*RESULTADO É 'DT#1998-02-13-14:20' *)
```

Exemplos em ST:

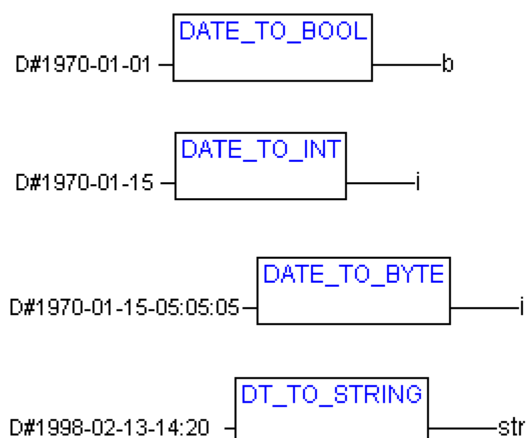
```
B :=DATE_TO_BOOL(D#1970-01-01);   (*RESULTADO É FALSE *)

I :=DATE_TO_INT(D#1970-01-15);    (*RESULTADO É 29952 *)

BYT :=DT_TO_BYTE(DT#1970-01-15-05:05:05); (*RESULTADO É 129 *)

STR:=DT_TO_STRING(DT#1998-02-13-14:20);  (*RESULTADO É 'DT#1998-02-13-14:20' *)
```

Exemplos em FBD:



Conversões STRING_TO

Conversão da variável tipo STRING para um tipo diferente:

O operando da variável tipo STRING deve conter um valor que seja válido no tipo de variável do dispositivo, caso contrário, o resultado será 0.

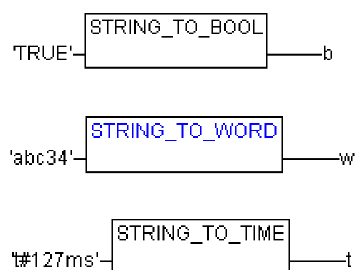
Exemplos em IL:

```
LD 'TRUE'
STRING_TO_BOOL
ST B            (*RESULTADO É TRUE *)
LD 'ABC34'
STRING_TO_WORD
ST W            (*RESULTADO É 0 *)
LD 'T#127MS'
STRING_TO_TIME
ST T(*RESULTADO É T#127MS *)
```

Exemplos em ST:

```
B: =STRING_TO_BOOL('TRUE');        (*RESULTADO É TRUE *)
W: =STRING_TO_WORD('ABC34');       (*RESULTADO É 0 *)
T: =STRING_TO_TIME('T#127MS');     (* RESULTADO É T#127MS *)
```

Exemplos em FBD:



TRUNC

Conversão de REAL para INT. A parte de número inteiro do valor será usada.

Ao executar uma conversão de um tipo maior para um tipo menor, arrisca-se perder algumas informações.

Exemplos em IL:

```
LD 2.7
TRUNC
GE %MW8
```

Exemplos em ST:

```
I:=TRUNC (1.9); (* RESULTADO É 1 *)
I: =TRUNC (-1.4); (* RESULTADO É -1 *)
```

Operadores numéricos

ABS

Retorna o valor absoluto de um número. ABS (-2) é igual a 2.

As seguintes combinações de tipo para variáveis de entrada e saída são possíveis:

IN	OUT
INT	INT, REAL, WORD, DWORD, DINT
REAL	REAL
BYTE	INT, REAL, BYTE, WORD, DWORD, DINT
WORD	INT, REAL, WORD, DWORD, DINT
DWORD	REAL, DWORD, DINT
SINT	REAL
USINT	REAL
UINT	INT, REAL, WORD, DWORD, DINT, UDINT, UINT
DINT	REAL, DWORD, DINT
UDINT	REAL, DWORD, DINT, UDINT

Tabela 7-1. Combinações de tipo para variáveis de entrada e saída em ABS

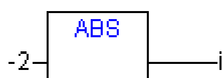
Exemplo em IL:

```
LD  -2
ABS
ST  I  (* RESULTADO É 2 *)
```

Exemplo em ST:

```
I:=ABS (-2);
```

Exemplo em FBD:



SQRT

Retorna a raiz quadrada de um número.

IN pode ser do tipo BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. OUT deve ser do tipo REAL.

Exemplo em IL:

```
 LD  16
SQRT
ST  Q (* RESULTADO É 4 *)
```

Exemplo em ST:

```
Q:=SQRT(16);
```

Exemplo em FBD:



LN

Retorna o logaritmo natural de um número.

IN pode ser do tipo BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. OUT deve ser do tipo REAL.

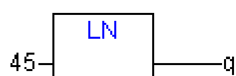
Exemplo em IL:

```
LD  45
LN
ST Q (* RESULTADO É 3.80666 *)
```

Exemplo em ST:

```
Q:=LN (45);
```

Exemplo em FBD:



LOG

Retorna o logaritmo de um número na base10.

IN pode ser de tipo BYTE, WORD, DWORD, INT, DINT, a TRUE, SINT, USINT, UINT, UDINT, OUT deve ser do tipo REAL.

Exemplo em IL:

```
LD  314.5
LOG
```

ST Q (* RESULTADO É 2.49762 *)

Exemplo em ST:

Q:=LOG (314.5);

Exemplo em FBD:



EXP

Retorna a função exponencial.

IN pode ser de tipo BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. OUT deve ser do tipo REAL.

Exemplo em IL:

LD 2

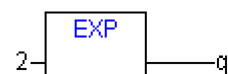
EXP

ST Q (* RESULTADO É 7.389056099 *)

Exemplo em ST:

Q:=EXP (2);

Exemplo em FBD:



SIN

Retorna o seno de um número.

O valor de entrada IN é calculado em arco minutos. Pode ser de tipo BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. OUT deve ser do tipo REAL.

Exemplo em IL:

LD 0.5

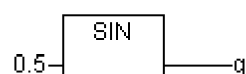
SIN

ST Q (* RESULTADO É 0.479426 *)

Exemplo em ST:

Q:=SIN (0.5);

Exemplo em FBD:



COS

Retorna o cosseno de um número. O resultado é calculado em minutos.

IN pode ser de tipo BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. OUT deve ser do tipo REAL.

Exemplo em IL:

```
LD 0.5

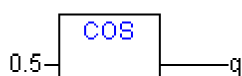
COS

ST Q (* RESULTADO É 0.877583 *)
```

Exemplo em ST:

```
Q:=COS (0.5);
```

Exemplo em FBD:



TAN

Retorna a tangente de um número. O valor é calculado em arco minutos.

IN pode ser de tipo BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. OUT deve ser do tipo REAL.

Exemplo em IL:

```
LD 0.5

TAN

ST Q (* RESULTADO É 0.546302 *)
```

Exemplo em ST:

```
Q:=TAN (0.5);
```

Exemplo em FBD:



ASIN

Retorna o arco seno de um número (função inversa do seno).

IN pode ser de tipo BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. OUT deve ser do tipo REAL.

Exemplo em IL:

```
LD 0.5

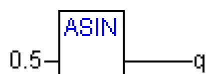
ASIN

ST Q (* RESULTADO É 0.523599 *)
```

Exemplo em ST:

```
Q:=ASIN (0.5);
```

Exemplo em FBD:



ACOS

Retorna o arco cosseno de um número (função inversa do cosseno). O valor é calculado em arco minutos.

IN pode ser de tipo BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. OUT deve ser do tipo REAL.

Exemplo em IL:

```
LD 0.5

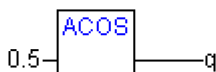
ACOS

ST Q  (* RESULTADO É 1.0472 *)
```

Exemplo em ST:

```
Q:=ACOS (0.5);
```

Exemplo em FBD:



ATAN

Retorna o arco tangente (função inversa da tangente).

IN pode ser de tipo BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. O resultado OUT é calculado em arco minutos e deve ser do tipo REAL.

Exemplo em IL:

```
LD 0.5

ATAN

ST Q(* RESULTADO É 0.463648 *)
```

Exemplo em ST:

```
Q:=ATAN (0.5);
```

Exemplo em FBD:



EXPT

Exponenciação de uma variável com outra variável:

$OUT = IN1^{IN2}$.

IN1 e IN2 podem ser de tipo BYTE, WORD, DWORD, INT, DINT, REAL, SINT, USINT, UINT, UDINT. OUT deve ser do tipo REAL.

Exemplo em IL:

```
LD 7

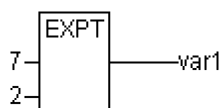
EXPT 2

ST VAR1 (* RESULTADO É 49 *)
```

Exemplo em ST:

```
VAR1:= EXPT (7,2);
```

Exemplo em FBD:



Operadores de inicialização

Operador INI

O operador INI pode ser usado para inicializar variáveis retentivas que são fornecidas por uma instância do bloco funcional utilizada na POU.

O operador deve ser atribuído a uma variável booleana.

Sintaxe: <Variável booleana > := INI(<Instância FB, TRUE|FALSE)

Se o segundo parâmetro do operador é definido como TRUE, todas variáveis retentivas definidas no bloco funcional FB serão inicializadas.

Exemplo em ST:

fbinst é a instância do bloco funcional fb, em que é definida uma variável retentiva retvar.

Declaração em POU:

```
FBINST: FB;

BBOOL;
```

Parte de implementação:

```
B:= IN I(FBINST, TRUE);

IVAR:=FBINST.RETVAR  (* => RETVAR É INICIALIZADA *)
```

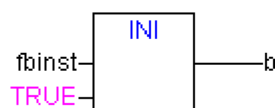
Exemplo de chamada do operador em IL:

```
LD FBINST

INI TRUE

ST B
```

Exemplo de chamada do operador em FBD:



Apêndice B: operandos no MasterTool IEC

No MasterTool IEC constantes, variáveis, endereços e, possivelmente, chamadas de função podem aparecer como operandos.

Constantes

Constantes booleanas

Constantes BOOL são os valores lógicos TRUE e FALSE.

Constantes de tempo

Constantes de TEMPO podem ser declaradas no MasterTool IEC. Elas são geralmente utilizadas para operar o temporizador na biblioteca padrão. Uma constante de TEMPO sempre é composta de uma inicial "t" ou "T" (ou "time" ou "TIME" enunciados) e um sinal numérico "#".

Ela é seguida pela declaração de tempo atual que pode incluir dias (identificados por "d"), horas (identificadas por "h"), minutos (identificados por "m"), segundos (identificados por "s") e milissegundos (identificados por "ms"). Por favor, note que as entradas de tempo devem ser fornecidas nesta ordem, de acordo com o tamanho da grandeza (d antes de h antes de m antes de s antes de ms), mas não é solicitada a inclusão de todos os incrementos do tempo.

Exemplos de constantes de TEMPO corretas em uma atribuição ST:

TIME1 := T#14ms;	
TIME1 := T#100S12ms;	(* O maior componente pode ser autorizado a exceder seu limite *)
TIME1 := t#12h34m15s;	

Tabela 7-2. Exemplos de constantes de TEMPO corretas em uma atribuição ST

O seguinte estaria incorreto:

TIME1 := t#5m68s;	(*limite excedido em um componente inferior*)
TIME1 := 15ms;	(*Está faltando T#*)
TIME1 := t#4ms13d;	(*Ordem incorreta de entradas*)

Tabela 7-3. Atribuições incorretas em constantes de tempo

Constantes de hora do dia

Use este tipo de constantes para armazenar as horas do dia. Uma declaração de TIME_OF_DAY começa com "tod#", "TOD#", "TIME_OF_DAY#" ou "time_of_day#" seguido de um tempo com o formato horas: minutos: segundos. Pode-se inserir segundos como números reais ou pode-se inserir frações de um segundo.

Exemplos:

TIME_OF_DAY#15:36:30.123

TOD#00:00:00

Consultar também: Tipos de dados de tempo

Constantes de data

Estas constantes podem ser usadas para inserir datas. Uma constante de DATA é declarada começando com um "d", "D", "DATE" ou "date" seguido de "#". Então, pode-se introduzir qualquer data com formato ano-mês-dia.

Exemplos:

DATE#1996-05-06

D#1972-03-29

Consultar também: Tipos de dados de tempo

Constantes de data e hora

Constantes de data e hora do dia também podem ser combinadas para formar as chamadas Constantes DATE_AND_TIME. Constantes DATE_AND_TIME começam com "dt#", "DT#", "DATE_AND_TIME#" ou "date_and_time#". Coloque um hífen após a data seguida pela hora.

Exemplos:

DATE_AND_TIME#1996-05-06-15:36:30

DT#1972-03-29-00:00:00

Consultar também: Tipos de dados de tempo

Constantes numéricas

Valores numéricos podem aparecer como números binários, números octais, números decimais e números hexadecimais. Se um valor inteiro não for um número decimal, deve-se escrever sua base seguida pelo sinal numérico (#) na frente da constante inteira. Os valores para os números 10-15 números hexadecimais estarão representados como sempre pelas letras A-F.

Pode-se incluir o caractere sublinhado dentro do número.

Exemplos:

14	(número decimal)
2#1001_0011	(número binário)
8#67	(número octal)
16#A	(número hexadecimal)

Tabela 7-4. Exemplos de constantes numéricas

Esses valores numéricos podem ser dos tipos de variáveis: BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL ou LREAL.

Conversões implícitas de tipos de variáveis maiores para "menores" não são permitidas. Isto significa que uma variável DINT simplesmente não pode ser utilizada como uma variável INT. Deve-se usar a conversão de tipo.

Constantes de cadeia de caracteres

Uma string é uma seqüência de caracteres. Constantes STRING são precedidas e seguidas por aspas simples. Também podem ser inseridos espaços em branco e caracteres especiais (*umlauts*, por exemplo). Eles serão tratados tal como todos os outros caracteres.

Em seqüências de caracteres, a combinação do sinal de dólar ($) seguida de dois números hexadecimais é interpretada como uma representação hexadecimal do código de caractere de oito bits. Além disso, a combinação de dois caracteres que começam com o sinal de dólar é interpretada como mostrado a seguir quando aparecem em uma seqüência de caracteres:

$$	Sinais de dólar
$'	Aspa única
$L ou $l	Avanço de linha
$N ou $n	Nova linha
$P ou $p	Avanço de página
$R ou $r	Quebra de linha
$T ou $t	Tab

Tabela 7-5. Constantes STRING

Exemplos:

'W1WÜß?'

'DUO ALTUS'

':-)'

Literais digitados

Basicamente, nas constantes IEC, o menor tipo de dados possível será usado. Se outro tipo de dados deve ser utilizado, isso pode ser obtido com a ajuda de literais digitados sem a necessidade de declarar explicitamente as constantes. Para isso, a constante será fornecida com um prefixo que determina o tipo.

Isto é escrito do seguinte modo: <Tipo>#<Literal>

<Tipo> especifica o tipo de dados desejado; entradas possíveis são: BOOL, SINT, USINT, BYTE, INT, UINT, WORD, DINT, UDINT, DWORD, REAL, LREAL. O tipo deve ser escrito em letras maiúsculas.

<Literal> especifica a constante. Os dados inseridos devem se ajustar ao tipo de dados especificado sob <Tipo>.

Exemplo:

var1:=DINT#34;

Se a constante não pode ser convertida para o tipo do dispositivo sem perda de dados, é emitida uma mensagem de erro:

Literais digitados podem ser usados sempre que constantes normais puderem ser utilizadas.

Variáveis

Variáveis podem ser declaradas quer localmente na parte declarações de uma POU quer em uma lista de variáveis globais.

Considerar: em um projeto, pode-se definir uma variável local que tenha o mesmo nome de uma variável global. Neste caso, dentro de uma POU a variável definida localmente será utilizada. Não é permitido, no entanto, nomear duas variáveis globais de forma idêntica. Por exemplo, ocorrerá um erro de compilador, se tiver definida uma variável "var1" na Configuração do CP e também em uma lista de variáveis globais.

O identificador de variável não pode conter qualquer espaços em branco ou caracteres especiais, não pode ser declarado mais de uma vez e não pode ser igual a qualquer uma das palavras-chave. Capitalização não é reconhecida, o que significa que VAR1, Var1 e var1 são a mesma variável. O caractere sublinhado é reconhecido em identificadores (por exemplo, "A_BCD" e "AB_CD" são considerados dois identificadores diferentes). Um identificador não pode ter mais de um caractere sublinhado em uma linha. O comprimento do identificador, bem como a parte com o seu significado, são ilimitados.

Variáveis podem ser usadas em qualquer lugar que o tipo declarado lhes permita.

Pode-se acessar variáveis disponíveis através do Assistente de entrada.

Memórias do sistema

Memórias de sistema são variáveis implicitamente declaradas que são diferentes em cada CP específico. Para descobrir quais memórias de sistema estão disponíveis no seu sistema, use o comando **'Inserir' 'Operando'**. Uma caixa de diálogo do Assistente de Entrada aparece e seleciona-se a categoria **Variáveis do Sistema**.

Acessando variáveis para matizes, estruturas e POUs.

Componentes da matriz bidimensional podem ser acessados usando a seguinte sintaxe:

<Nomedocampo>[Indice1,Indice2]

Variáveis de estrutura podem ser acessadas usando a seguinte sintaxe:

<Nomedaestrutura>.<Nomedavariável>

Variáveis de blocos funcionais e programas podem ser acessadas usando a seguinte sintaxe:

<Nomedoblocofuncional>.<Nomedavariável>

Endereçando bits em variáveis

Em variáveis inteiras bits individuais podem ser acessados. Para isso, o índice do bit a ser endereçado é anexado à variável, separado por um ponto. O índice do bit pode ser concedido por qualquer constante. A indexação é baseada em 0.

Considerar: acesso a bit em Variáveis diretas não é permitido.

Exemplo:

```
A : INT;
B : BOOL;
...
A.2 := B;
```

O terceiro bit da variável a será definido para o valor da variável b.

Se o índice for maior que a largura em bit da variável, a seguinte mensagem de erro é emitida: Índice '<n>' fora do intervalo válido para a variável '<var>'!

O endereçamento de bits é possível com os seguintes tipos de variáveis: SINT, INT, DINT, USINT, UINT, UDINT, BYTE, WORD, DWORD.

Se o tipo de variável não permiti-lo, é emitida a seguinte mensagem de erro: "Tipo de dados inválidos, '<tipo>' para indexação direta"

Um acesso de bits não deve ser atribuído a uma variável VAR_IN_OUT!

Acesso a bit por meio de uma constante global:

Ao declarar uma constante global, que define o índice de bit, pode-se usar essa constante para um acesso a bit.

Considerar: a opção do projeto 'Substituir constantes' (categoria Compilar) deve estar ativada!

Veja nos exemplos a seguir o acesso a bit em uma variável e em uma variável de estrutura:

Declaração na lista de variáveis globais para ambos os exemplos:

A variável “ENABLE” define quais bits devem ser acessados:

```
VAR_GLOBAL CONSTANT
ENABLE:INT:=2;
END_VAR
```

Exemplo 1 (acesso a bit em uma variável inteira):

Declaração na POU:

```
VAR
XXX:INT;
END_VAR
```

Acesso a bit:

xxx.enable:=TRUE; -> o terceiro bit na variável xxx será definido TRUE

Exemplo 2 (acesso a bit em um componente de estrutura inteiro):

Declaração da estrutura stru1:

```
TYPE STRU1 :
STRUCT
BVAR:BOOL;
RVAR:REAL;
WVAR:WORD;
```

```
{BITACESS ENABLE 42 'START DRIVE'}
END_STRUCT
END_TYPE
```

Declaração na POU:

```
VAR
X:STRU1;
END_VAR
```

Acesso a bit:

```
X.ENABLE:=TRUE;
```

Este procedimento provocará SET TRUE no bit 42 na variável x. Uma vez que bvar tem 8 bits e rvar tem 32 bits, o acesso a bit será feito sobre o segundo bit da variável wvar, que como resultado, irá obter o valor 4.

> Atenção: se uma variável, que faz um acesso a bit em uma variável de estrutura com a ajuda de uma constante global, deve ser exibida corretamente no Assistente de Entrada, no monitoramento na janela de declaração e na função intellisense, por favor, use o pragma {bitaccess}, como mostrado no exemplo. Então, adicionalmente será indicada a constante global, além da variável de estrutura respectiva durante a monitoração na janela de declaração:

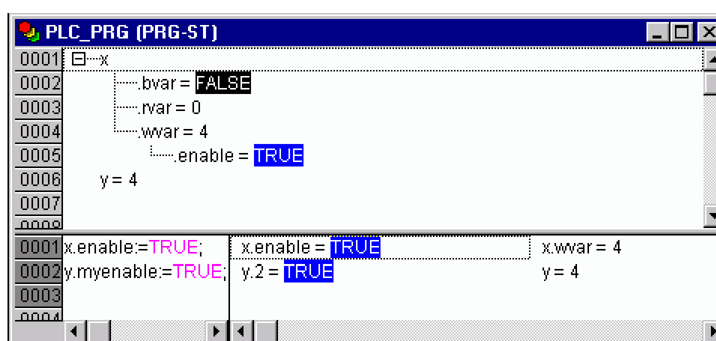


Figura 7-1. Ilustração do exemplo anterior

Endereços

Endereço

A exibição direta de locais individuais de memória é feita através da utilização de seqüências de caracteres especiais. Estas seqüências são uma concatenação do sinal de porcentagem "%", um prefixo de intervalo, um prefixo para o tamanho e um ou mais números naturais, separados por espaços em branco.

Os seguintes prefixos de intervalo são suportados:

I	Entrada
Q	Saída
M	Local da memória

Tabela 7-6. Prefixos de intervalo

Os seguintes prefixos de tamanho são suportados:

X	bit
Nenhum	bit
B	Byte (8 Bits)
W	Word (16 Bits)
D	Double word (32

	Bits)

Tabela 7-7. Prefixos de tamanho

Exemplos:

%QX7.5 e %Q7.5 Bit de saída 7.5

%IW215 Palavra de entrada 215

%QB7 Byte de saída7

%MD48 Palavra dupla na posição de memória 48

%IW2.5.7.1 Dependendo da configuração do CP

ivar AT %IW0 : WORD; Exemplo de uma declaração de variável incluindo uma atribuição de endereço

A configuração atual do CP para o programa determina se um endereço é válido ou não.

Nota: valores booleanos serão alocados por byte, se não for especificado nenhum endereço de bit simples explícito. Exemplo: uma alteração no valor de varbool1 em %QW0 afeta o intervalo de QX0.0 a QX0.7.

Consultar também no Apêndice A: Operadores de endereço

Locais de memória

Pode-se usar qualquer tamanho suportado para acessar locais de memória.

Por exemplo, o endereço %MD48 endereçaria os números de bytes 192, 193, 194 e 195 na área de local de memória (48 * 4 = 192). O número do primeiro byte é 0.

Pode-se acessar palavras, bytes e mesmo bits da mesma forma: o endereço %MX5.0 permite acessar o primeiro bit na quinta palavra (bits são geralmente salvos por palavra).

Consultar também no Apêndice A: Operadores de endereço

Funções

Em ST, uma chamada de função pode aparecer com um operando.

Exemplo:

Resultado:= Fct (7) + 3;

Função TIME ()

Esta função retorna o tempo (baseado em milissegundos) que passou desde que o sistema foi iniciado.

O tipo de dados é TIME.

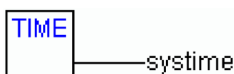
Exemplo em IL:

```
TIME
ST SYSTIME (* RESULTADO E.G.: T#35M11S342MS *)
```

Exemplo em ST:

```
SYSTIME: =TIME();
```

Exemplo em FBD:



Apêndice C: tipos de dados no MasterTool IEC

Tipos de Dados Padrão

Pode-se usar tipos de dados padrão e tipos de dados definidos pelo usuário quando fizer a programação. Cada identificador é atribuído a um tipo de dados que determina quanto espaço de memória será reservado e que tipo de valores ele armazena.

Booleanos

Variáveis do tipo BOOL podem determinar os valores TRUE e FALSE. Serão reservados 8 bits de espaço de memória.

Consultar no Apêndice B: Operandos no MasterTool IEC, Constantes BOOL.

Inteiros

BYTE, **WORD**, **DWORD**, **SINT**, **USINT**, **INT**, **UINT**, **DINT** e **UDINT** são todos tipos de dados inteiros.

Cada um dos diferentes tipos numéricos cobre um intervalo diferente de valores. As seguintes limitações de intervalo se aplicam aos tipos de dados inteiros:

Tipo	Limite Inferior	Limite Superior	Espaço de memória
BYTE	0	255	8 bit
WORD	0	65535	16 bit
DWORD	0	4294967295	32 bit
SINT:	-128	127	8 bit
USINT:	0	255	8 bit
INT:	-32768	32767	16 bit
UINT:	0	65535	16 bit
DINT:	-2147483648	2147483647	32 bit
UDINT:	0	4294967295	32 bit

Tabela 7-8. Tipos de Dados Inteiros

Em conseqüência, quando tipos maiores são convertidos para tipos menores, informações podem ser perdidas.

Consultar no Apêndice B: Operandos no MasterTool IEC, Constantes Numéricas.

Reais (consultar disponibilidade)

REAL e LREAL são chamados de tipos de pontos flutuante. Eles devem representar os números racionais. Para REAL, 32 bits de espaço de memória são reservados e, para LREAL, 64 bits.

Valores válidos para REAL: 1.175494351e-38 a 3.402823466e+38

Valores válidos para LREAL: 2.2250738585072014e-308 a 1.7976931348623158e+308

Consultar também: Constantes REAL-/ LREAL (verificar disponibilidade)

Cadeia de caracteres (STRING)

Uma variável de tipo **STRING** pode conter qualquer seqüência de caracteres. O tamanho da entrada na declaração determina quanto espaço de memória deve ser reservado para a variável. Refere-se ao número de caracteres na string e pode ser colocado entre parênteses ou colchetes quadrados. Se nenhuma especificação de tamanho for fornecida, o tamanho padrão de 80 caracteres será usado.

O comprimento de uma String basicamente não está limitada no MasterTool IEC, mas funções de string somente podem processar strings de 1 a 255 caracteres!

Exemplo de uma declaração de String com 35caracteres:

```
STR:STRING(35):='THIS IS A STRING';
```

Consultar no Apêndice B: Operandos no MasterTool IEC, Constantes String

Tipos de Dados de Tempo

Os tipos de dados **TIME**, **TIME_OF_DAY** (abr. TOD), **DATE**, e **DATE_AND_TIME** (abr. DT) são manipulados internamente como DWORD.

O tempo é dado em milissegundos, em TIME e TOD, tempo em TOD começa às 12:00 a.m.

O tempo é dado em segundos em DATE e DT começando em Janeiro 1, 1970, às 12:00 a.m.

Veja a seguir os formatos de dados de tempo usados para atribuir valores para constantes de tempo:

Constantes de tempo

Sempre constituído por uma inicial "t" ou "T" (ou "time" ou "TIME" enunciados) e um sinal numérico "#".

Este é seguido pela declaração de tempo atual que pode incluir dias (identificados por "d"), horas (identificadas por "h"), minutos (identificados por "m"), segundos (identificados por "s") e milissegundos (identificados por "ms"). Considerar que as entradas de tempo devem ser dadas nesta ordem, de acordo com comprimento (d antes de h antes de m antes de s antes de ms), mas não é solicitada a inclusão de todos os incrementos do tempo.

Valor máximo: 49d17h2m47s295ms (4194967295 ms)

Exemplos de constantes de TEMPO corretos em uma atribuição ST:

TIME 1 := T#14ms;	
TIME 1 := T#100S12ms;	(*O maior componente pode exceder seu limite*)
TIME 1 := t#12h34m15s;	

Tabela 7-9. Exemplos de constantes de TEMPO

O que segue estaria incorreto:

TIME 1 := t#5m68s;	(*o menor componente excedeu o limite*)
TIME 1 := 15ms;	(*está faltando T#*)
TIME 1 := t#4ms13d;	(*ordem incorreta das entradas*)

Tabela 7-10. Atribuições incorretas para constantes de tempo

Constantes de data

Uma constante de data começa com um "d", D", "date" ou "DATE" seguido de"#". Pode-se então introduzir qualquer data com o formato Ano-Mês-Dia. Valores possíveis: 1970-00-00 até 2106-02-06.

Exemplos:

```
DATE#1996-05-06
D#1972-03-29
```

*Constantes **TIME_OF_DAY**, para armazenar as horas do dia*

Começam com tod#, TOD#, TIME_OF_DAY# ou "time_of_day#" seguido de um tempo com o formato: horas: minutos: segundos. Segundos podem ser inscritos como números reais ou pode-se inserir frações de um segundo. Valores possíveis: 00:00:00 - 23:59:59.999.

Exemplos:

```
TIME_OF_DAY#15:36:30.123

TOD#00:00:00
```

*Constantes **DATE_AND_TIME**, combinação de data e hora do dia*

Começam com "dt#", "DT#", "DATE_AND_TIME#" ou "date_and_time#". Coloque um hífen após a data seguida do tempo. Valores possíveis: 1970-00-00-00:00:00 to 2106-02-06-06:28:15.

Exemplos:

```
DATE_AND_TIME#1996-05-06-15:36:30

DT#1972-03-29-00:00:00
```

Tipos de Dados Definidos pelo Usuário

Matriz (Array)

Um, dois e três campos dimensionais (matrizes) são suportados como tipos de dados elementares. Matrizes podem ser definidas tanto na parte de declaração de uma POU quanto nas listas de variáveis globais. Um máximo de 9 dimensões podem resultar do aninhamento de matrizes ("ARRAY[0..2] OF ARRAY[0..3] OF …").

Sintaxe:

<Nome_campo>:ARRAY [<ll1>..<ul1>,<ll2>..<ul2>] OF <tipo elem.>.

ll1, ll2, ll3 identificam o limite inferior do campo de intervalo; ul1, ul2 e ul3 identificam o limite superior. Os valores limite devem ser inteiros e devem coincidir com o intervalo de valor DINT.

Exemplo:

Card_game: ARRAY [1..13, 1..4] OF INT;

Inicializando matrizes

Exemplo de uma inicialização completa de uma matriz:

```
ARR1 : ARRAY [1..5] OF INT := 1,2,3,4,5;

ARR2: ARRAY [1..2, 3..4] OF INT := 1,3(7); (* SHORT FOR 1,7,7,7 *)

ARR3 : ARRAY [1..2,2..3,3..4] OF INT := 2(0),4(4),2,3;

   (* SHORT FOR 0, 0,4,4,4,4,2,3 *)
```

Exemplo de uma inicialização de uma estrutura de matriz:

```
TYPE STRUCT1

STRUCT

P1: INT;

P2: INT;

P3: DWORD;

END_STRUCT

ARRAY [1..3 ] OF STRUCT1:= (P1:=1,P2:=10,P3:=4723),(P1:=2,P2:=0,P3:=299),

(P1:= 14,P2:=5,P3:=112);
```

Exemplo de uma inicialização parcial de uma matriz:

```
ARR1: ARRAY [ 1..10] OF INT := 1,2;
```

Elementos aos quais nenhum valor é pré-atribuído são inicializados com o valor inicial padrão do tipo básico. No exemplo acima, os elementos anarray [6] a anarray [10], por conseguinte, são inicializados com 0.

Acessando componentes de matriz:

Componentes de matriz são acessados em uma matriz bidimensional usando a seguinte sintaxe:

<Nome_campo>[Index1,Index2]

Exemplo:

```
CARD_GAME [9,2]
```

> Nota: ao definir uma função no projeto com o nome CheckBounds, pode-se usá-la para verificação de estouros de intervalo no projeto (ver capítulo "Conceitos associados ao MasterTool IEC', 'Componentes de um projeto', 'Função').

Função CheckBounds

Ao definir uma função no projeto com o nome CheckBounds, pode-se verificar automaticamente erros fora do intervalo em matrizes. O nome da função é fixo e só pode ter essa designação.

Exemplo para a função CheckBounds:

```
FUNCTION CHECKBOUNDS : DINT
VAR_INPUT
INDEX, LOWER, UPPER: DINT;
END_VAR
IF INDEX < LOWER THEN
 CHECKBOUNDS: = LOWER;
ELSIF INDEX > UPPER THEN
CHECKBOUNDS:= UPPER;
ELSE CHECKBOUNDS: = INDEX;
END_IF
```

O seguinte exemplo de programa para testar a função CheckBounds excede os limites de uma matriz definida. A função CheckBounds permite que o valor TRUE a ser atribuído, não ao local A [10], mas para o limite de intervalo ainda válido A [7] acima dele. Com a função CheckBounds, referências fora dos limites da matriz, por conseguinte, podem ser corrigidas.

Programa de teste para a função CheckBounds:

```
PROGRAM PLC_PRG
VAR
 A: ARRAY[0..7] OF BOOL;
 B: INT:=10;
END_VAR
A[B]:=TRUE;
```

> Atenção: a função CheckBounds fornecida pela biblioteca Check.Lib é apenas uma solução de amostra! Antes de usar esse módulo de biblioteca, verifique se a função está funcionando no seu sentido, ou implemente uma função apropriada diretamente como uma POU no projeto.

Ponteiro

Endereços de variáveis ou blocos funcionais são salvos em ponteiros enquanto um programa está em execução

Declarações de ponteiro têm a seguinte sintaxe:

<Identificador>: POINTER TO <Tipo de dados/Bloco funcional>;

Um ponteiro pode apontar para qualquer tipo de dados ou bloco funcional, mesmo os tipos definidos pelo usuário.

A função do Operador de Endereço ADR é atribuir o endereço de uma variável ou bloco funcional ao ponteiro.

Um ponteiro pode ser de-referenciado, adicionando o operador de conteúdo "^" após o identificador de ponteiro.

Por favor, observe: um ponteiro é incrementado de forma binária! Pode-se incrementá-lo, da maneira habitual como é feito no compilador C, usando a instrução p = p+SIZEOF(p^);.

Exemplo:

```
PT: POINTER TO INT;

VAR_INT1: INT: = 5;

VAR_INT2: INT;

PT:= ADR(VAR_INT1);

VAR_INT2:= PT^; (* VAR_INT2 É AGORA 5 *)
```

Verificando os acessos do ponteiro no tempo de execução

Para verificar os acessos do ponteiro durante o tempo de execução pode-se criar funções de verificação, que serão chamadas automaticamente antes de cada acesso no endereço de um ponteiro. Para esse efeito a função respectiva deve estar disponível no projeto, diretamente, ou através de uma biblioteca. As seguintes funções são suportadas:

- Função CheckPointer: verifica se o endereço atualmente armazenado no ponteiro está dentro do intervalo válido de memória,
- CheckPointerAligned: executa a funcionalidade do CheckPointer e, além disso, verifica o alinhamento de memória.

As funções devem ter exatamente os nomes mencionados. Elas retornam o endereço que é usado para de-referênciar o ponteiro, assim, na melhor das hipóteses aquele que tenha sido transferido como o primeiro parâmetro de entrada (dwAddress no exemplo mostrado abaixo).

Consultar o seguinte exemplo de uma função CheckPointerAligned, na qual os parâmetros de entrada são processados. Os nomes de parâmetros são exemplos também. Uma função CheckPointer deve ter a mesma aparência, exceto que não pode haver nenhum parâmetro para a granularidade do acesso de ponteiro:

```
FUNCTION CHECKPOINTERALIGNED : DWORD      (* O TIPO DE DADOS DA FUNÇÃO
                                           (VALOR DE RETORNO) DEVE SER O
                                           MESMO USADO PARA PONTEIROS
                                           NO ATUAL SISTEMA DO
                                           DISPOSITIVO DEFINIDO NO
                                           MOMENTO; OU SEJA, DWORD PARA
                                           SISTEMAS USANDO PONTEIROS DE
                                           32 BITS, WORD PARA SISTEMAS
                                           USANDO PONTEIROS DE 16 BITS *)

VAR_INPUT

 DWADDRESS : DWORD;                       (* ENDEREÇO DO DISPOSITIVO DO
                                           PONTEIRO; O TIPO DE DADOS DEVE
                                           SER O MESMO QUE USADO PARA
                                           PONTEIROS NO SISTEMA DO
                                           DISPOSITIVO DEFINIDO NO
                                           MOMENTO, VEJA ACIMA: VALOR DE
```

	RETORNO*)
ISIZE : DINT;	(* TAMANHO DO ACESSO DE PONTEIRO; OS TIPOS DE DADOS DEVEM SER INTEIRO-COMPATÍVEIS E DEVEM COBRIR O TAMANHO DE DADOS POTENCIAL MÁXIMO ARMAZENADO NO ENDEREÇO DE PONTEIRO *)
IGRAN : DINT;	(*! NÃO DEVEM SER UTILIZADOS NAS FUNÇÕES DE CHECKPOINTER!: GRANULARIDADE DO ACESSO, POR EXEMPLO, "2", SE INT FOR O MENOR TIPO DE DADOS NÃO-ESTRUTURADOS USADO EM DETERMINADO ENDEREÇO; O TIPO DE DADOS DEVERÁ SER INTEIRO-COMPATÍVEL*)
BWRITE: BOOL;	(* TIPO DE ACESSO: LEITURA OU ESCRITA; TRUE = ACESSO DE LEITURA; O TIPO DE DADOS DEVE SER BOOL *)

END_VAR

Se estiverem presentes as funções CheckPointer e CheckPointerAligned no projeto, será chamada a função CheckPointerAligned.

Enumeração

Enumeração é um tipo de dados definido pelo usuário que é constituído por um número de constantes string. Essas constantes são denominadas valores de enumeração.

Valores de enumeração são reconhecidos em todos os domínios do projeto, mesmo se forem declaradas dentro de uma POU. É melhor criar suas enumerações como objetos no Organizador de objetos sob o cartão de registro Tipos de dados. Eles começam com a palavra-chave TYPE e terminam com END_TYPE.

Sintaxe:

```
TYPE < IDENTIFICADOR>:(<ENUM_0> ,<ENUM_1>, ...,<ENUM_N>);

END_TYPE
```

Uma variável do tipo <Identificador> pode assumir um dos valores de enumeração e será inicializada com o primeiro. Esses valores são compatíveis com números inteiros, o que significa que pode-se executar operações com eles como faria com INT. Pode-se atribuir um número x para a variável. Se os valores de enumeração não são inicializados, a contagem começará com 0. Ao inicializar, certifique os valores iniciais estão aumentando. A validade do número será revista no momento que é executado.

Exemplo:

```
TYPE TRAFFIC_SIGNAL: (RED, YELLOW, GREEN:=10); (*O VALOR INICIAL PARA CADA UMA DAS CORES
É RED 0, YELLOW 1, GREEN 10 *)

END_TYPE

TRAFFIC_SIGNAL1 : TRAFFIC_SIGNAL;

TRAFFIC_SIGNAL1:=0; (* O VALOR DO SINAL DE TRÂNSITO É RED*)

FOR I:= RED TO GREEN DO
```

```
 I := I + 1;
END_FOR;
```

O mesmo valor de enumeração não pode ser utilizado duas vezes dentro de uma enumeração, ou de todas as enumerações utilizadas na mesma POU.

Exemplo:

```
TRAFFIC_SIGNAL: (RED, YELLOW, GREEN);
COLOR: (BLUE, WHITE, RED);
```

Erro: vermelho não pode ser utilizado para TRAFFIC_SIGNAL e COLOR

Estruturas

Estruturas são criadas como objetos no Organizador de objetos no cartão de registro Tipos de dados. Elas começam com as palavras-chave TYPE e STRUCT e terminam com END_STRUCT e END_TYPE.

A sintaxe de declarações de estrutura é a seguinte:

```
TYPE < NOME DA ESTRUTURA >:
STRUCT
<DECLARAÇÃO DE VARIÁVEIS 1>
.
.
<DECLARAÇÃO DE VARIÁVEIS N>
END_STRUCT
END_TYPE
```

<Nome da estrutura> é um tipo que é reconhecido em todo o projeto e pode ser usado como um tipo de dados padrão.

Estruturas intertravadas são permitidas. A única restrição é que as variáveis não podem ser colocadas em endereços (a declaração AT não é permitida!).

Exemplo de uma definição de estrutura denominada Polygonline:

```
TYPE POLYGONLINE:
STRUCT
 START:ARRAY [1..2] OF INT;
 POINT1:ARRAY [1..2] OF INT;
 POINT2:ARRAY [1..2] OF INT;
 POINT3:ARRAY [1..2] OF INT;
 POINT4:ARRAY [1..2] OF INT;
 END:ARRAY [1..2] OF INT;
END_STRUCT
END_TYPE
```

Exemplo para a inicialização de uma estrutura:

Poly_1:polygonline := (Start:=3,3, Point1 =5,2, Point2:=7,3, Point3:=8,5, Point4:=5,7, End := 3,5);

Inicializações com variáveis não são possíveis. Veja um exemplo da inicialização de uma matriz de uma estrutura em 'Matrizes'.

Acesso a componentes da estrutura: pode-se ter acesso aos componentes da estrutura utilizando a seguinte sintaxe...

<Nome_Estrutura>.<Nome do componente>

Portanto, para o exemplo acima referido da estrutura 'polygonline', pode-se acessar o componente 'start' por Poly_1.Start.

Referências

Pode-se utilizar o tipo de dados de referência definidas pelo usuário para criar um nome alternativo para uma variável, constante ou bloco funcional.

Crie suas referências como objetos no Organizador de Objetos sob o cartão de registro Tipos de Dados. Eles começam com a palavra-chave TYPE e terminam com END_TYPE.

Sintaxe:

TYPE <Identificador>:<Termo de atribuição>;

END_TYPE

Exemplo:

```
TYPE MESSAGE:STRING[50];

END_TYPE;
```

Tipos de sub-intervalo

Um tipo sub-intervalo é um tipo cujo intervalo de valores é apenas um sub-conjunto do tipo básico. A declaração pode ser efetuada no registro de tipos de dados, mas uma variável também pode ser declarada diretamente com um tipo de sub-intervalo:

Sintaxe de declaração no registro 'Tipos de dados':

TYPE <Nome> : <TipoInt> (<ug>.<og>) END_TYPE;

<Nome>	Deve ser um identificador IEC válido,
<TipoInt>	é um dos tipos de dados SINT, USINT, INT, UINT, DINT, UDINT, BYTE, WORD, DWORD (LINT, ULINT, LWORD).
<ug>	É uma constante que deve compatíveis com o tipo básico e que define o limite inferior dos tipos de intervalo. O limite inferior próprio está incluído neste intervalo.
<og>	É uma constante que deve ser compatível com o tipo básico e define o limite superior dos tipos de intervalo. O limite superior próprio está incluído neste tipo de base.

Tabela 7-11. Tipos de sub-intervalo

Exemplos:

```
TYPE

SUBINT: INT (-4095..4095);

END_TYPE
```

Declaração direta de uma variável com o tipo sub-intervalo:

```
VAR
 I : INT (-4095..4095);
 UI : UINT (0..10000);
END_VAR
```

Se uma constante é atribuída a um tipo sub-intervalo (na declaração ou na implementação) ela não cai neste intervalo (por exemplo, 1:= 5000), uma mensagem de erro é emitida.

Para verificar a observância dos limites de intervalo em tempo de execução, é preciso introduzir as funções CheckRangeSigned ou CheckRangeUnsigned. Nestas, podem ser capturadas violações de limite pelo método e meios adequados (por exemplo, o valor pode estar fora do limite ou um erro de sinalizador pode ser definido.). Elas são chamadas implicitamente logo que uma variável é escrita (pertencente a um tipo de sub-intervalo construído a partir de um tipo com ou sem sinal).

Exemplos:

No caso de uma variável pertencente a um tipo de sub-intervalo com sinal (como i, acima), a função CheckRangeSigned é chamada; poderia ser programado do seguinte modo para limitar um valor no intervalo permitido:

```
FUNCTION CHECKRANGESIGNED : DINT
VAR_INPUT
 VALUE, LOWER, UPPER: DINT;
END_VAR
IF (VALUE < LOWER) THEN
 CHECKRANGESIGNED := LOWER;
ELSIF(VALUE > UPPER) THEN
 CHECKRANGESIGNED := UPPER;
ELSE
 CHECKRANGESIGNED := VALUE;
END_IF
```

Ao chamar a função automaticamente, o nome de função CheckRangeSigned é obrigatório, como é a especificação da interface: valor de retorno e três parâmetros do tipo DINT

Quando chamado, a função é parametrizada do seguinte modo:

- value: o valor a ser atribuído para o tipo de intervalo
- lower: o limite inferior do intervalo
- upper: o limite superior do intervalo
- -Return value: (valor de retorno) este é o valor efetivamente atribuído ao tipo de intervalo

Uma atribuição i:=10*y implicitamente produz o seguinte neste exemplo:

i:= CheckRangeSigned(10*y, -4095, 4095);

Mesmo se, por exemplo, temos o valor 1000, então, i ainda tem apenas o valor 4095 após esta atribuição.

O mesmo se aplica a função CheckRangeUnsigned: o nome da função e a interface devem estar corretos.

FUNCTION CheckRangeUnsigned: UDINT

VAR_INPUT

value, lower, upper: UDINT;

END_VAR i: = CheckRangeSigned (10*y,-4095, 4095);

Importante: se nenhuma das funções CheckRangeSigned ou CheckRangeUnsigned estiver presente, nenhuma verificação dos tipos de sub-intervalo ocorre durante o tempo de execução! A variável i poderia, então, ter qualquer valor entre –32768 e 32767 a qualquer momento!

Atenção: se nenhuma das funções CheckRangeSigned ou CheckRangeUnsigned está presente como descrito acima, pode resultar um laço interminável se um tipo de sub-intervalo é usado em um laço FOR. Isso irá acontecer quando o intervalo para o laço for tão grande quanto ou maior que o intervalo do tipo de sub-intervalo!

Atenção: a função CheckRangeSigned fornecida com a biblioteca de Check.Lib é apenas uma solução de amostra! Antes de usar o módulo de biblioteca, verifique se a função está funcionando conforme solicitado para seu projeto, ou implemente uma função CheckRange adequada diretamente como uma POU no projeto.

Exemplo:

```
VAR

UI : UINT (0..10000);

END_VAR

FOR UI:=0 TO 10000 DO

...

END_FOR
```

O laço FOR nunca teminará, porque UI não pode ser maior do que 10000.

Também atente para a definição das funções CheckRange ao definir o valor incremental de um laço FOR!

Apêndice D: as bibliotecas no MasterTool IEC

Biblioteca STANDARD lib

Funções STRING

Por favor, observe: funções de string não são do tipo "segmento seguro"! Ao usar tarefas, funções de string só podem ser utilizadas em uma única tarefa. Se a mesma função é usada em tarefas diferentes, há um risco de sobrescrever.

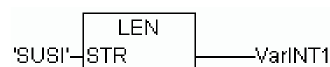
LEN

Retorna o comprimento de uma string. A entrada STR é do tipo STRING, o valor de retorno da função é do tipo INT.

Exemplo em IL:

```
LD 'SUSI'
LEN
ST VARINT1 (* RESULTADO É 4 *)
```

Exemplo em FBD:



Exemplo em ST:

```
VARSTRING1:= LEN ('SUSI');
```

LEFT

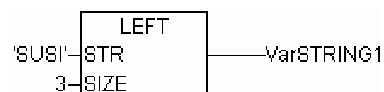
Left retorna a string inicial esquerda, para uma determinada string. A entrada STR é do tipo STRING, SIZE do tipo INT e o valor de retorno da função é do tipo STRING.

LEFT (STR, SIZE) significa: retorne o primeiro caractere SIZE da esquerda na string STR.

Exemplo em IL:

```
LD 'SUSI'
LEFT 3
ST   VARSTRING1 (* RESULTADO É 'SUS' *)
```

Exemplo em FBD:



Exemplo em ST:

```
VARSTRING1:= LEFT ('SUSI',3);
```

RIGHT

RIGHT retorna a string inicial direita para uma determinada string.

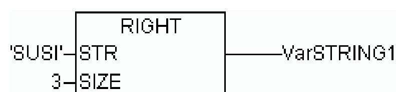
RIGHT (STR, SIZE) significa: pegue o primeiro caractere SIZE da direita na string STR.

A entrada STR é do tipo STRING, SIZE é do tipo INT, o valor de retorno da função é do tipo STRING.

Exemplo em IL:

```
LD 'SUSI'
RIGHT 3
ST    VARSTRING1 (* RESULT IS 'USI' *)
```

Exemplo em FBD:



Exemplo em ST:

```
VARSTRING1:= RIGHT ('SUSI',3);
```

MID

MID retorna uma string parcial de dentro de uma string.

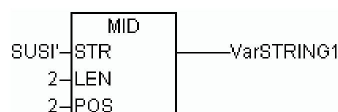
A entrada STR é do tipo STRING, LEN e POS do tipo INT, o valor de retorno da função é do tipo STRING.

MID (STR, LEN, POS) significa: recuperar LEN caracteres da string STR começando com o caractere na posição POS.

Exemplo em IL:

```
LD  'SUSI'
MID 2,2
ST  VARSTRING1 (* RESULT IS 'US' *)
```

Exemplo em FBD:



Exemplo em ST:

```
VARSTRING1:= MID ('SUSI', 2, 2);
```

CONCAT

Concatenação (combinação) de duas strings.

As variáveis de entrada STR1 e STR2 bem como o valor de retorno da função são do tipo STRING.

Exemplo em IL:

```
LD     'SUSI'
CONCAT 'WILLI'
ST     VARSTRING1 (* RESULT IS 'SUSIWILLI' *)
```

Exemplo em FBD:



Exemplo em ST:

```
VARSTRING1 := CONCAT ('SUSI','WILLI');
```

Considerar: a função CONCAT não funciona, se aninhada por mais de cinco níveis.

INSERT

INSERT insere uma string em outra string, em um ponto definido.

As variáveis de entrada STR1 e STR2 são do tipo STRING, POS é do tipo INT e o valor de retorno da função é do tipo STRING.

INSERT (STR1, STR2, POS) significa: inserir STR2 em STR1 após a posição POS.

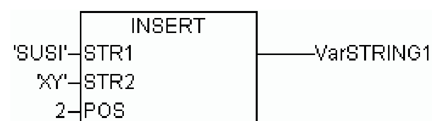
Exemplo em IL:

```
LD    'SUSI'
```

INSERT 'XY',2

ST   VARSTRING1 (* RESULT IS 'SUXYSI' *)

Exemplo em FBD:



Exemplo em ST:

VARSTRING1:= INSERT ('SUSI','XY',2);

DELETE

DELETE remove uma string parcial de uma string maior em uma posição definida.

A variável de entrada STR é do tipo STRING, LEN e POS são do tipo INT e o valor de retorno da função é do tipo STRING.

DELETE (STR, L, P) significa: excluir L caracteres de STR, começando com o caractere na posição P.

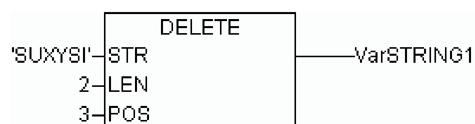
Exemplo em IL:

LD'SUXYSI'

DELETE 2,3

ST VAR1 (* RESULTADO É 'SUSI' *)

Exemplo em FBD:



Exemplo em ST:

VAR1:= DELETE ('SUXYSI',2,3);

REPLACE

REPLACE substitui uma string parcial de uma string maior em uma terceira string.

As variáveis de entrada STR1 e STR2 são do tipo STRING, LEN e POS são do tipo INT e o valor de retorno da função é do tipo STRING.

REPLACE (STR1, STR2, L, P) significa: substituir L caracteres da STR1 com STR2, começando com o caractere na posição P.

Exemplo em IL:

LD SUXYSI'

REPLACE 'K',2,2

ST VARSTRING1 (* RESULTADO É 'SKYSI' *)

Exemplo em FBD:



Exemplo em ST:

VARSTRING1:= REPLACE ('SUXYSI','K',2,2);

FIND

FIND procura por uma string parcial dentro de uma string.

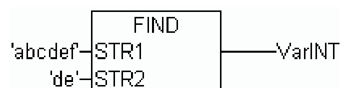
As variáveis de entrada STR1 e STR2 são do tipo STRING, e o valor de retorno da função é do tipo STRING.

FIND (STR1, STR2) significa: localizar a posição do primeiro caractere onde aparece STR2 na STR1 pela primeira vez. Se STR2 não for encontrado em STR1, então OUT: = 0.

Exemplo em IL:

```
LD   'ABCDEF'
FIND 'DE'
ST   VARINT1 (* RESULT IS '4' *)
```

Exemplo em FBD:



Exemplo em ST:

```
ARINT1:= FIND ('ABCDEF','DE');
```

Blocos funcionais biestáveis

SR

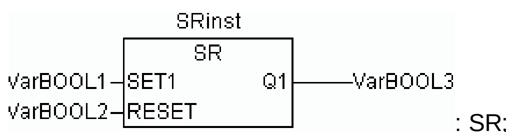
Fazendo blocos funcionais biestáveis dominantes:

Q1 = SR (SET1, RESET) means:

Q1 = (NOT RESET AND Q1) OR SET1

As variáveis de entrada SET1 e RESET bem como a variável de saída Q1 são tipo BOOL.

Exemplo de declaração:



: SR;

Exemplo em IL:

```
CAL SRINST (SET1 := VARBOOL1, RESET := VARBOOL2)
LD SRINST.Q1
ST VARBOOL3
```

Exemplo em FBD:

Exemplo em ST:

```
SRINST (SET1:= VARBOOL1, RESET: =VARBOOL2 );
VARBOOL3:= SRINST. Q1;
```

RS

Fazendo reset nos blocos funcionais biestáveis:

Q1 = RS (SET1, RESET) significa:

Q1 = NOT RESET1 AND (Q1 OR SET)

As variáveis de entrada SET e RESET1, bem como a variável de saída Q1 são do tipo BOOL.

Exemplo de declaração:

```
RSINST: RS ;
```

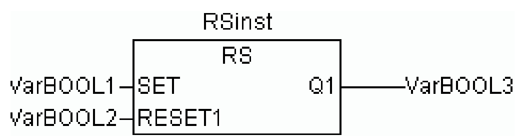
Exemplo em IL:

```
CAL RSINST (SET:= VARBOOL1,RESET1:=VARBOOL2)
```

LD RSINST.Q1

ST VARBOOL3

Exemplo em FBD:



Exemplo em ST:

RSINST (SET:= VARBOOL1 , RESET1:=VARBOOL2);

VARBOOL3:= RSINST.1 ;

SEMA

Um Semáforo por Software (passível de interrupção).

BUSY = SEMA (CLAIM, RELEASE) MEANS:

BUSY:= X;

IF CLAIM THEN X: =TRUE;

ELSE IF RELEASE THEN BUSY = FALSE; X: = FALSE;

END_IF

X é uma variável BOOL interna que é FALSE, quando é inicializada. As variáveis de entrada CLAIM e RELEASE, bem como a variável de saída BUSY são do tipo BOOL.

Se BUSY é TRUE quando SEMA é chamada, então isto significa que já foi atribuído um valor para SEMA (SEMA foi chamada com CLAIM = TRUE). Se BUSY for FALSO, SEMA não foi ainda chamada ou foi liberada (chamadas com RELEASE = TRUE).

Exemplo de declaração:

SEMAINST: SEMA ;

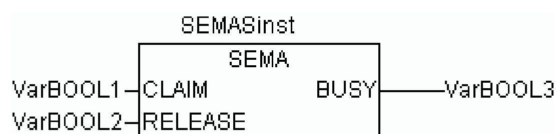
Exemplo em IL:

CAL SEMAINST (CLAIM:=VARBOOL1,RELEASE:=VARBOOL2)

LD SEMAINST.BUSY

ST VARBOOL3

Exemplo em FBD:



Exemplo em ST:

SEMAINST (CLAIM:= VARBOOL1 , RELEASE:=VARBOOL2);

VARBOOL3:= SEMAINSTBUSY;

Disparo
R_TRIG

O bloco funcional R_TRIG detecta uma borda de subida.

FUNCTION_BLOCK R_TRIG

VAR_INPUT

 CLK : BOOL;

END_VAR

VAR_OUTPUT

```
 Q : BOOL;
END_VAR
VAR
 M: BOOL := FALSE;
END_VAR
 Q: = CLK AND NOT M;
 M:= CLK;
```

A saída Q e a variável de ajuda M permanecem FALSE enquanto a variável de entrada CLK é FALSE. Assim que CLK retornar TRUE, Q primeiro retornará TRUE, depois M será definido como TRUE. Isto significa que a cada vez que a função é chamada, Q retornará FALSE até que CLK tenha sua borda de descida seguida por uma borda de subida.

Exemplo de declaração:

```
RTRIGINST: R_TRIG ;
```

Exemplo em IL:

```
CAL RTRIGINST (CLK:= VARBOOL1)
LD  RTRIGINST.Q
ST VARBOOL2
```

Exemplo em FBD:



Exemplo em ST:

```
RTRIGINST (CLK:= VARBOOL1);
VARBOOL2:= RTRIGINST.Q;
```

F_TRIG

O bloco funcional F_TRIG detecta uma borda de descida.

```
FUNCTION_BLOCK F_TRIG
VAR_INPUT
 CLK: BOOL;
END_VAR
VAR_OUTPUT
 Q: BOOL;
END_VAR
VAR
 M: BOOL := FALSE;
END_VAR
 Q: = NOT CLK AND NOT M;
 M:= NOT CLK;
```

A saída Q e a variável de ajuda M permanecerão FALSE enquanto a variável de entrada CLK retornar TRUE. Assim que CLK retornar FALSE. Q primeiro retornará TRUE, depois M será definido como TRUE. Isto significa que a cada vez que a função é chamada, Q retornará FALSE até que CLK tenha sua borda de subida seguida por uma borda de descida.

Exemplo de declaração:

```
FTRIGINST: F_TRIG ;
```

Exemplo em IL:

```
CAL FTRIGINST (CLK := VARBOOL1)
LD FTRIGINST.Q
ST VARBOOL2
```

Exemplo em FBD:



Exemplo em ST:

```
FTRIGINST (CLK:= VARBOOL1);
VARBOOL2:= F
```

Contadores

CTU

Bloco funcional de incremento.

As variáveis de entrada CU e RESET, bem como a variável de saída Q são do tipo BOOL, a variável de entrada PV e a de saída CV são do tipo WORD.

A variável de contador CV será inicializada com 0 se RESET for TRUE. Se CU tiver uma borda de subida de FALSE para TRUE, CV será aumentada em 1. Q retornará TRUE quando CV é maior ou igual ao limite superior de PV.

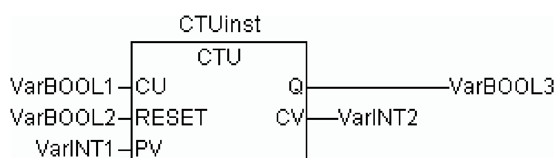
Exemplo de declaração:

```
CTUINST: CTU ;
```

Exemplo em IL:

```
CAL CTUINST (CU:= VARBOOL1, RESET := VARBOOL2, PV := VARINT1)
LD   CTUINST.Q
ST   VARBOOL3
LD   CTUINST.CV
ST VARINT2
```

Exemplo em FBD:



Exemplo em ST:

```
CTUINST(CU:= VARBOOL1, RESET:=VARBOOL2 , PV:= VARINT1);
VARBOOL3:= CTUINST.Q ;
VARINT2:= CTUINST.CV;
```

CTD

Bloco Funcional de decremento.

As variáveis de entrada CD e LOAD, assim como a variável de saída Q, são do tipo BOOL, a variável de entrada PV e a de saída CV são do tipo WORD.

Quando LOAD_ é TRUE, a variável de contador CV será inicializada com o limite superior de PV. Se CD tem uma borda de subida de FALSE para TRUE, CV decrementará em 1, enquanto CV for maior que 0 (isto é, não causa a queda do valor abaixo de 0).

Q retorna TRUE quando CV é igual a 0.

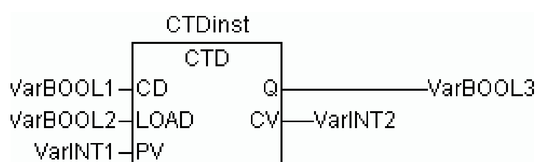
Exemplo de declaração:

CTDINST : CTD ;

Exemplo em IL:

```
CAL CTDINST (CD: = VARBOOL1, LOAD := VARBOOL2, PV := VARINT1)
LD CTDINST.Q
ST VARBOOL3
LD CTDINST. CV
ST VARINT2
```

Exemplo em FBD:



Exemplo em ST:

```
CTDINST (CD:= VARBOOL1, LOAD:=VARBOOL2 , PV:= VARINT1);
VARBOOL3:= CTDINST.Q;
VARINT2:= CTDINST.CV;
```

CTUD

Blocos funcionais de incremento/decremento.

As variáveis de entrada CU, CD, RESET e LOAD, assim como as variáveis de saída QU e QD, são do tipo BOOL, PV e CV são do tipo WORD.

Se RESET é válido, a variável de contador CV será inicializada com 0. Se LOAD é válido, CV será inicializado com PV.

Se CU tem uma borda de subida (FALSE para TRUE), CV incrementará em 1. Se CD em uma borda de subida (FALSE para TRUE), CV decrementará em 1, desde que isto é não cause a queda do valor abaixo de 0.

QU retorna TRUE quando CV tornar-se maior ou igual a PV.

QD retorna TRUE quando CV tornar-se igual a 0.

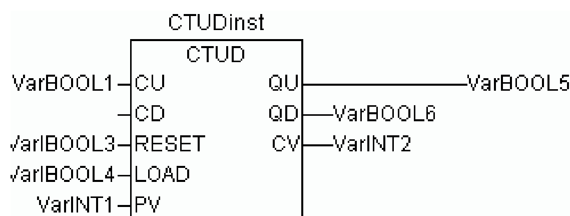
Exemplo de declaração:

CTUDINST: CUTD ;

Exemplo em IL:

```
CAL CTUDINST(CU:=VARBOOL1, RESET:=VARBOOL3, LOAD:=VARBOOL4, PV:=VARINT1)
LD   CTUDINST.Q
ST   VARBOOL5
LD   CTUDINST.QD
ST   VARBOOL5
LD   CTUINST.CV
ST VARINT2
```

Exemplo em FBD:



Exemplo em ST:

```
CTUDINST (CU := VARBOOL1, RESET := VARBOOL3, LOAD:=VARBOOL4 , PV:= VARINT1);
VARBOOL5:= CTUDINST.QU ;
VARBOOL6:= CTUDINST.QD ;
VARINT2:= CTUDINST.CV;
```

Temporizadores

TP

O bloco funcional TP é um disparador TP(IN, PT, Q, ET).

IN e PT são variáveis de entrada dos tipos BOOL e TIME respectivamente. Q e ET são variáveis de saída dos tipos BOOL e TIME, respectivamente. Se IN é FALSE, Q é FALSE e ET é 0.

Logo que IN se torne TRUE, o tempo irá começar a ser contado em milissegundos em ET até que seu valor seja igual a PT. Em seguida, ele permanecerá constante.

Q é TRUE assim que IN tornou-se TRUE e ET é igual ou inferior a PT. Caso contrário, é FALSE.

Q retorna um sinal para o período de tempo fornecido em PT.

Exibição gráfica da Seqüência de Tempo TP:

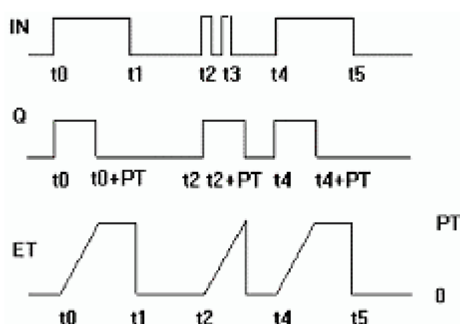


Figura 7-2. Exibição gráfica da Seqüência de Tempo TP

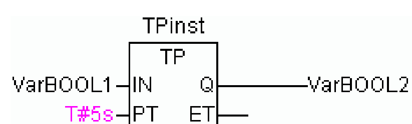
Exemplo de declaração:

```
TPINST: TP ;
```

Exemplo em IL:

```
CAL TPINST (IN: = VARBOOL1, PT := T#5S)
LD TPINST.Q
ST VARBOOL2
```

Exemplo em FBD:



Exemplo em ST:

```
TPINST (IN := VARBOOL1, PT:= T#5S);
VARBOOL2:=TPINST.Q;
```

TON

O bloco funcional Timer On Delay implementa um retardo na energização.

TON (IN, PT, Q, ET) significa:

IN e PT são variáveis de entrada dos tipos BOOL e TIME respectivamente. Q e ET são variáveis de saída dos tipos BOOL e TIME, respectivamente. Se IN é FALSE, Q é FALSE e ET é 0.

Logo que IN se torne TRUE, o tempo irá começar a ser contado em milissegundos em ET até que seu valor seja igual a PT. Em seguida, ele permanecerá constante.

Q é TRUE quando IN é TRUE e ET é igual a PT.Caso contrário, é FALSE.

Assim, Q tem uma borda de subida quando o tempo indicado em PT em milissegundos esgotou-se.

Exibição gráfica do comportamento de TON ao longo do tempo.

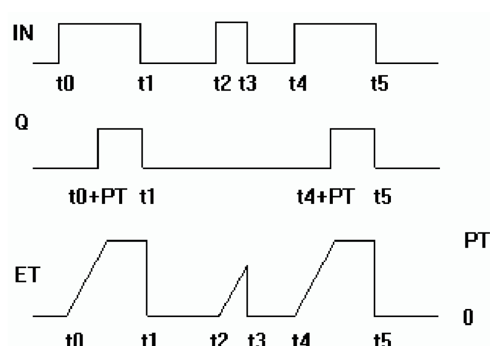


Figura 7-3. Exibição gráfica do comportamento de TON

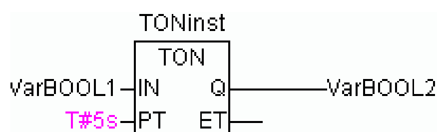
Exemplo de declaração:

```
TONINST : TON ;
```

Exemplo em IL:

```
CAL TONINST(IN := VARBOOL1, PT := T#5S)
LD  TONINST.Q
ST  VARBOOL2
```

Exemplo em FBD:



Exemplo em ST:

```
TONINST (IN := VARBOOL1, PT:= T#5S);
```

TOF

O bloco funcional TOF implementa um retardo na desenergização.

TOF(IN, PT, Q, ET) significa:

IN e PT são variáveis de entrada dos tipos BOOL e TIME respectivamente. Q e E são variáveis de saída dos tipos BOOL e TIME, respectivamente. Se IN é TRUE, as saídas são TRUE, respectivamente 0.

Logo que IN se torne FALSO, em ET o tempo irá começar a ser contado em milissegundos até que seu valor seja igual a PT. Em seguida, ele permanecerá constante.

Q é FALSE quando IN é FALSE e ET é igual a PT.Caso contrário, é TRUE.

Assim, Q tem uma borda de descida quando o tempo indicado em PT em milissegundos esgotou-se.

Exibição gráfica do comportamento de TOF ao longo do tempo.

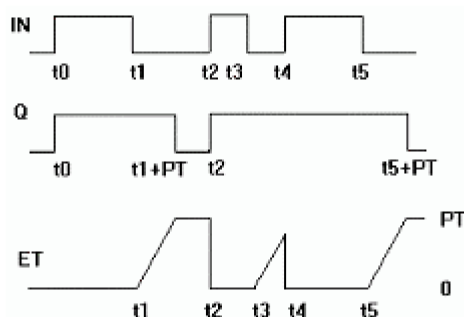


Figura 7-4. Exibição gráfica do comportamento de TOF

Exemplo de declaração:

TOFINST: TOF ;

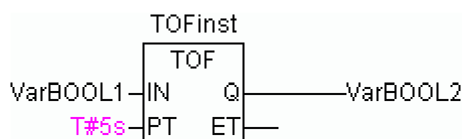
Exemplo em IL:

CAL TOFINST (IN := VARBOOL1, PT := T#5S)

LD TOFINST.Q

ST VARBOOL2

EXEMPLO EM FBD:



Exemplo em ST:

TOFINST (IN: = VARBOOL1, PT:= T#5S);

VARBOOL2:= TOFINST. Q;

RTC

O bloco funcional relógio de tempo de execução retorna, começando por um determinado período, a data atual e a hora.

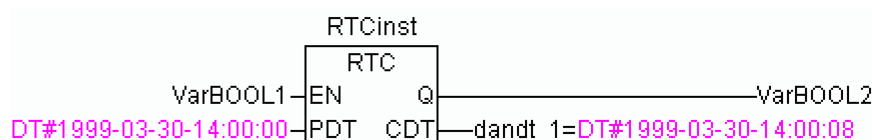


Figura 7-5. Bloco funcional relógio de tempo de execução

RTC(EN, PDT, Q, CDT) significa:

EN e PDT são variáveis do tipo BOOL e DATE_AND_TIME respectivamente. Quando em é FALSE, as variáveis Q e CDT são FALSE, respectivamente DT#1970-01-01-00:00:00.

Assim que EN tornar-se TRUE, o tempo de PDT é definido, é contabilizado em segundos e retornado em CDT, enquanto EN for TRUE (ver exemplo na imagem acima). Logo que EN é redefinido como FALSE, CDT é redefinido para o valor inicial DT # 1970-01-01-00: 00: 00. Por favor, observe que o tempo em PDT só é definido por uma borda de subida.

Biblioteca UTIL.lib

Esta biblioteca contém uma coleção adicional de vários blocos, os quais podem ser usados para conversão BCD, funções bit/byte, funções auxiliares matemáticas, como controlador, geradores de sinal, manipuladores de funções e para processamento de valores analógicos.

Conversão BCD

Um byte no formato BCD contém números inteiros entre 0 e 99. Quatro bits são usados para cada casa decimal. A décima casa decimal é armazenada nos bits de 4-7. Assim, o formato BCD é semelhante ao da apresentação hexadecimal, com a simples diferença de que apenas os valores entre 0 e 99 podem ser armazenados em um byte BCD, enquanto um byte hexadecimal atinge de 0 a FF.

Um exemplo: o inteiro 51 deve ser convertido para formato BCD. Em binário, 5 é 0101e 1 é 0001, o que torna o byte BCD 01010001e corresponde ao valor de $51=81.

BCD_TO_INT

Esta função converte um byte no formato BCD em um valor INT:

O valor de entrada da função é do tipo BYTE e a saída é do tipo INT.

Quando um byte que não está no formato BCD precisar ser convertido, a saída é-1.

Exemplos em ST:

I := BCD_TO_INT(73); (* RESULTADO É 49 *)

K := BCD_TO_INT(151); (* RESULTADO É 97 *)

L := BCD_TO_INT(15); (* SAÍDA -1, PORQUE NÃO ESTÁ NO FORMATO BCD *)

INT_TO_BCD_

Esta função converte um valor INTEIRO em um byte no formato BCD:

O valor de entrada da função é do tipo INT e a saída é do tipo BYTE.

O número 255 será fornecido quando um valor INTEIRO deve ser convertido e não puder ser convertido em um byte BCD.

Exemplos em ST:

I := INT_TO_BCD(49); (* RESULTADO É 73 *)

K := BCD_TO_INT(97); (* RESULTADO É 151 *)

L := BCD_TO_INT(100); (* ERRO! SAÍDA: 255 *)

Funções Bit-Byte

EXTRACT

Entradas para esta função são DWORD X, bem como um BYTE N. A saída é um valor BOOL, que contém o conteúdo do enésimo bit da entrada X, segundo o qual a função começa a contar do bit zero.

Exemplos em ST:

FLAG := EXTRACT(X:=81, N:=4); (* Resultado: TRUE, porque 81 é binário 1010001, portanto o quarto bit é 1 *)

FLAG: = EXTRACT (X:=33, N:=0); (* Resultado: TRUE, porque 33 é binário 100001, portanto o bit zero é 1 *)PACK

Esta função é capaz de entregar de volta oito bits de entrada B0, B1,..., B7 do tipo BOOL como um BYTE.

O bloco funcional UNPACK está intimamente relacionado com a esta função.

PACK

Esta função é capaz de retornar oito bits de entrada B0, B1, ..., B7 do tipo BOOL como um BYTE.

O bloco funcional UNPACK está intimamente relacionado a esta função.

PUTBIT

A entrada para esta função consiste de uma DWORD X, um BYTE N e um valor booleano B.

PUTBIT seta o enésimo bit de X sobre o valor de B, segundo o qual ele começa a contagem a partir do bit zero.

Exemplo em ST:

A: = 38; (* 100110 binário *)

B: =PUTBIT(A,4,TRUE); (* Resultado: 54 = 2 # 110110 *)

C: =PUTBIT (A,1,FALSE); (* Resultado: 36 = 2 # 100100 *)

UNPACK

UNPACK converte a entrada B do tipo BYTE em 8 variáveis de saída B0,..., B7 do tipo BOOL, e é o contrário PACK.

Exemplo em FBD:

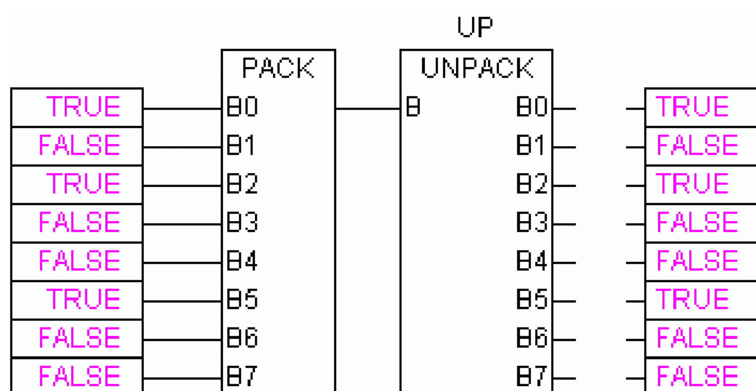


Figura 7-6. Bloco funcional UNPACK

Funções matemáticas auxiliares

DERIVATIVE

Este bloco funcional aproximadamente determina a derivação local.

O valor da função é entregue como uma variável REAL usando IN. TM contém o tempo decorrido em ms em um DWORD e a entrada de RESET do tipo BOOL permite ao bloco funcional iniciar um novo através a entrega do valor TRUE.

A saída OUT é do tipo REAL.

Para obter o melhor resultado possível, a DERIVAÇÃO aproxima usando os quatro últimos valores, para manter os erros que são produzidos por imprecisões nos parâmetros de entrada tão baixos quanto possível.

Bloco Funcional em FBD:

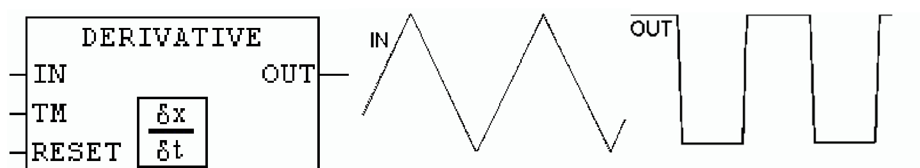


Figura 7-7. Operação do bloco funcional DERIVATIVE

INTEGRAL

Este bloco funcional determina aproximadamente a integral da função.

De forma análoga à DERIVAÇÃO, o valor da função é entregue como uma variável REAL usando IN. TM contém o tempo decorrido em ms em uma DWORD e a entrada de RESET do tipo BOOL permite ao bloco funcional iniciar de novo com o valor TRUE.

A saída OUT é do tipo REAL.

A integral é aproximada por duas funções de passo. A média destas é entregue como a integral aproximada.

Bloco funcional em FBD (por exemplo, integração de uma função linear):

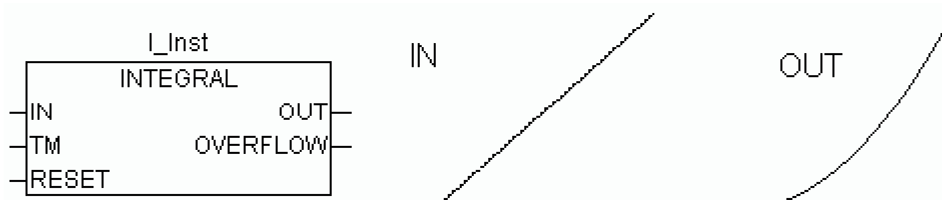


Figura 7-8. Bloco funcional INTEGRAL

LIN_TRAFO

Este bloco funcional transforma um valor REAL, o qual está em um intervalo de valor definido por um valor limite inferior e superior,em um valor REAL, o qual, por sua vez, está proporcionalmente em outro intervalo, também definido por um limite inferior e superior. A seguinte equação é a base da conversão:

(IN - IN_MIN) : (IN_MAX - IN) = (OUT - OUT_MIN) : (OUT_MAX - OUT)

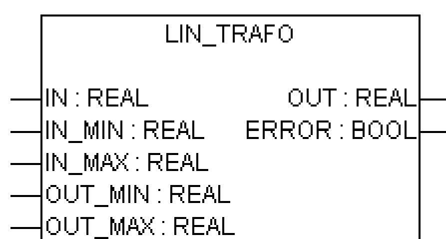


Figura 7-9. Bloco funcional LIN_TRAFO

Variáveis de entrada:

Variável	Tipo de Dados	Descrição
IN	REAL	Valor de entrada
IN_MIN	REAL	Limite inferior do intervalo de valores de entrada
IN_MAX	REAL	Limite superior do intervalo de valores de entrada
OUT_MIN	REAL	Limite inferior do intervalo de valores de saída
OUT_MAX	REAL	Limite de superior do intervalo de valores de saída

Tabela 7-12. Variáveis de entrada (LIN_TRAFO)

Variáveis de saída:

Variável	Tipo dados	Descrição
OUT	REAL	Valor de saída
ERROR	BOOL	Erro ocorreu: TRUE, se IN_MIN = IN_MAX, ou se IN está fora do intervalo de valor de entrada especificado.

Tabela 7-13. Variáveis de saída (LIN_TRAFO)

Exemplo de aplicação:

Um sensor de temperatura fornece valores em volt (entrada IN). Estes devem ser convertidos em valores de temperatura em graus Celsius (saída OUT). O intervalo de valores de entrada (volt) é definido pelos limites IN_MIN = 0 e IN_MAX = 10. O intervalo de valor de saída (grau Celsius) é definido pelos limites OUT_MIN = -20 e a OUT_MAX = 40.

Assim, para uma entrada de 5 V, resultará uma temperatura de 10 graus C.

STATISTICS_INT

Este bloco funcional calcula alguns valores estatísticos padrão.

A entrada IN é do tipo INT. Todos os valores são inicializados quando a entrada booleana RESET é TRUE.

A saída MN contém o mínimo, MX o valor máximo de IN. AVG descreve a média, que é o valor esperado de IN. Todas as saídas três são do tipo INT:

Bloco em FBD:



Figura 7-10. Bloco funcional STATISTICS_INT

STATISTICS_REAL

Este bloco funcional corresponde a STATISTICS_INT, exceto que a entrada IN é do tipo REAL, assim como as saídas MN, MX, AVG.

VARIANCE

VARIANCE calcula a variância em relação aos valores inseridos.

A entrada IN é do tipo REAL, RESET é do tipo BOOL e a saída OUT é novamente do tipo REAL.

Este bloco calcula a variância dos valores inseridos. VARIANCE pode ser redefinida com RESET = TRUE.

O desvio padrão facilmente pode ser calculado como a raiz quadrada da variância.

Controladores

PD

A biblioteca util.lib fornece o seguinte bloco funcional de controle PD:

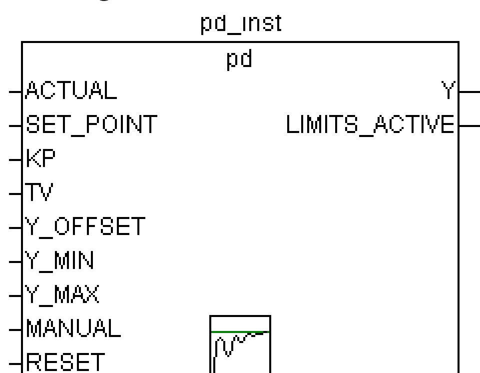


Figura 7-11. Bloco funcional PD

Entradas do bloco funcional:

Variáveis	Tipo de Dados	Descrição
ACTUAL	REAL	Valor atual da variável
SET_POINT	REAL	Valor desejado, da variável de comando
KP	REAL	Coeficiente de proporcionalidade, ganho de unidade das partes P
TV	REAL	Tempo de ação derivada, ganho de unidade das partes D em segundos, por exemplo, "0,5" para 500 ms.
Y_MANUAL	REAL	Define o valor da saída Y, em caso de MANUAL = TRUE
Y_OFFSET	REAL	Offset para a variável manipulada Y
Y_MIN, Y_MAX	REAL	Limites inferiores e superiores para a variável Y manipulada. Se Y exceder esses limites, a saída LIMITS_ACTIVE será definida como TRUE e Y será mantida dentro do intervalo prescrito. Este controle só funcionará se Y_MIN < Y_MAX.
MANUAL	BOOL	Se TRUE, a operação manual será ativa, ou seja, o valor manipulado será definido por Y_MANUAL.
RESET	BOOL	TRUE redefine o controlador; durante a reinicialização Y = Y_OFFSET.

Tabela 7-14. Bloco funcional PD (entradas)

Saídas do bloco funcional:

Variáveis	Tipo de Dados	Descrição
Y	REAL	Valor manipulado, calculado pelo bloco funcional (ver abaixo)
LIMITS_ACTIVE	BOOL	TRUE indica que Y excedeu os limites fornecidos (Y_MIN, Y_MAX).

Tabela 7-15. Bloco funcional PD (saídas)

Y _OFFSET, Y_MIN e _MAX são usados para a transformação da variável manipulada dentro de um intervalo prescrito.

MANUAL pode ser usado para alternar o acionamento da operação manual. RESET serve para zerar o controlador.

Em operações normais (MANUAL = RESET = LIMITS_ACTIVE = FALSE) o controlador calcula o e erro e as diferenças de SET_POINT – ACTUAL geram a derivação em relação ao tempo.

A saída, ou seja, a variável Y manipulada é calculada da seguinte forma:

Y = KP · (Δ + TV δΔ/δt) + Y_OFFSET onde Δ = SET_POINT-ACTUAL

Então, além da parte P, também a alteração atual do erro de controlador (parte D) influencia a variável manipulada.

Além disso, Y limita-se em um intervalo prescrito por Y_MIN e Y_MAX. Se Y exceder estes limites, LIMITS_ACTIVE será TRUE. Se não é desejada limitação da variável manipulada, Y_MIN e Y_MAX têm de ser definidas como 0.

Enquanto MANUAL = TRUE, Y_MANUAL será escrita como Y.

Um controle "P" pode ser implementado facilmente através da configuração TV = 0.

PID

A biblioteca util.lib fornece o seguinte bloco funcional de controle PID:

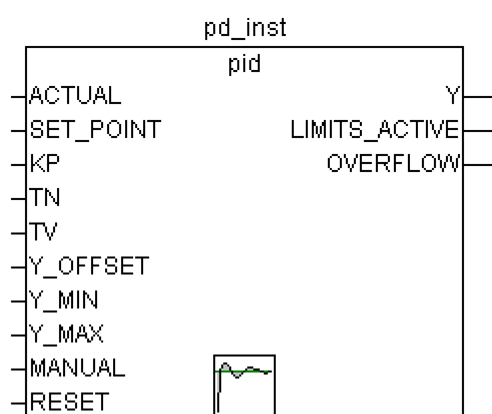


Figura 7-12. Bloco funcional PID

Ao contrário do controlador PD, este bloco funcional contém uma entradas adicional REAL TN para reajustar o tempo em ms (por exemplo, "0,5" de 500 ms).

Entradas do bloco funcional:

Variável	Tipo de Dados	Descrição
ACTUAL	REAL	Valor atual da variável controlada
SET_POINT	REAL	Valor desejado, variável de comando
KP	REAL	Coeficiente de proporcionalidade, ganho de unidade da parte P
TN	REAL	Redefinir o tempo, ganho de unidade recíproco da parte I; fornecido em segundos, por exemplo, "0,5" para 500 ms.
TV	REAL	Tempo de ação derivada, ganho de unidade da parte D em segundos, por exemplo, "0,5" para 500 ms.
Y_MANUAL	REAL	Define valor da saída Y, em caso de MANUAL = TRUE
Y_OFFSET	REAL	Offset para a variável manipulada Y
Y_MIN, Y_MAX	REAL	Limites. Superiores e inferiores para a variável manipulada Y. Se Y exceder estes limites, a saída LIMITS_ACTIVE será definido como TRUE e Y será mantida dentro do intervalo prescrito. Este controle só funcionará se Y_MIN < Y_MAX.
MANUAL	BOOL	Se TRUE, manual de operação vai estar ativo, ou seja, o valor manipulado será definido por Y_MANUAL.

RESET	BOOL	TRUE redefine o controlador; durante a reinicialização Y = Y_OFFSET.

Tabela 7-16. Bloco funcional PID (entradas)

Saídas do bloco funcional:

Variáveis	Tipo de Dados	Descrição
Y	REAL	Valor manipulado, calculado pelo bloco funcional (ver abaixo)
LIMITS_ACTIVE	BOOL	TRUE indica que Y excedeu os limites fornecidos (Y_MIN, Y_MAX).
OVERFLOW	BOOL	TRUE indica um estouro (veja abaixo).

Tabela 7-17. Bloco funcional PID (saídas)

Y _OFFSET, Y_MIN e _MAX são usados para a transformação da variável manipulada dentro de um intervalo prescrito.

MANUAL pode ser usado para alternar o acionamento da operação manual. RESET serve para zerar o controlador.

Em operações normais (MANUAL = RESET = LIMITS_ACTIVE = FALSE) o controlador calcula o e erro e as diferenças de SET_POINT – ACTUAL, geram a derivação em relação ao tempo.

A saída, ou seja, a variável Y manipulada , diferentemente do controlador PD contém uma parte adicional inteira e é calculada da seguinte forma:

$$Y = KP \cdot (\Delta + 1/TN \int e dt + TV \, \delta\Delta/\delta t) + Y_OFFSET$$

Então, além da parte P, também a alteração atual do erro de controlador (parte D) e o histórico do erro de controlador (parte I) influencia a variável manipulada.

O controlador de PID pode ser facilmente convertido em um controlador PI definindo TV = 0.

Devido à parte adicional integral, um estouro pode ocorrer devido à parametrização incorreta do controlador, se a integral do erro D se tornar muito grande. Por conseguinte, por razões de segurança um booleano de saída OVERFLOW está presente, que neste caso teria o valor TRUE. Isto só acontecerá se o sistema de controle fique instável devido à parametrização incorreta. Ao mesmo tempo, o controlador será suspenso e só irá ser ativado novamente por re-inicialização.

PID_FIXCYCLE

A biblioteca Util.lib fornece o bloco funcional controlador PID_FIXCYCLE:

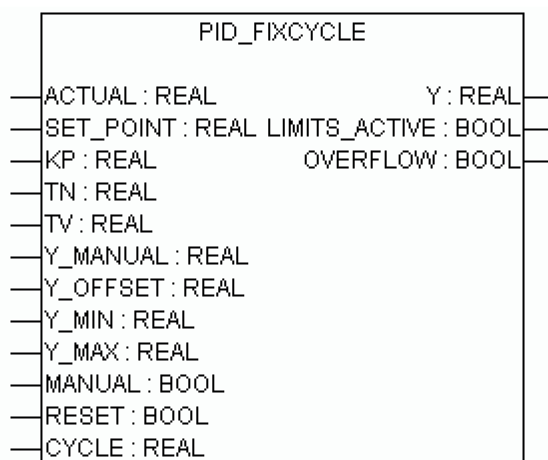


Figura 7-13. Controlador PID_FIXCYCLE

A funcionalidade deste bloco funcional corresponde a do controlador PID, com a exceção de que o tempo de ciclo não é medido automaticamente por uma função interna, mas é definido pelo ciclo de entrada (em segundos).

Geradores de sinal

BLINK

O bloco funcional BLINK gera um sinal pulsante. A entrada consiste em ENABLE do tipo BOOL, bem como TIMELOW e TIMEHIGH do tipo TIME. A saída OUT é do tipo BOOL.

Se ENABLE é definido como TRUE, BLINK começa, ligando a saída (TRUE) para o período de tempo TIMEHIGH e depois a desligando (FALSE) para o período de tempo TIMELOW.

Quando ENABLE é redefinido como FALSE, a saída OUT não será alterada, ou seja, não serão gerados mais pulsos. Se explicitamente também deseja-se obter, FALSE, quando ENABLE é redefinido como FALSE, pode-se usar o "OUT AND ENABLE" (ou seja, adicionar uma caixa AND com o parâmetro ENABLE) na saída.

Exemplo in CFC:

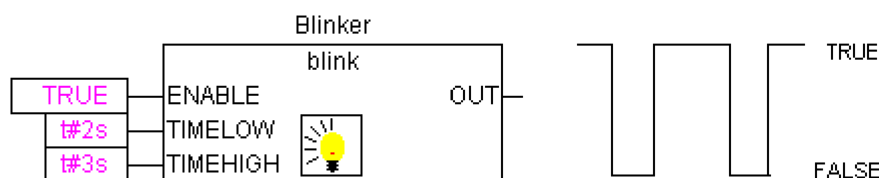


Figura 7-14. Exemplo de BLINK em CFC

FREQ_MEASURE

Este bloco funcional mede a freqüência (média) (Hz) de um sinal de entrada booleano. Pode-se especificar durante quantos períodos deve ser média. Um período é o tempo entre as duas bordas crescentes do sinal de entrada.

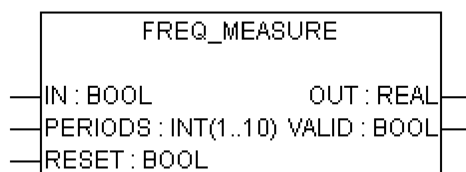


Figura 7-15. Bloco funcional FREQ_MEASURE

Variáveis de entrada:

IN (BOOL) Sinal de entrada

PERIODS (INT) Número de períodos. Valores possíveis: 1 a 10

RESET (BOOL) Reset de todos os parâmetros para 0

Variáveis de saída:

OUT (REAL) f(resultante) [Hz]

VALID (BOOL) FALSE até que a primeira medição tenha terminado, ou se o período > 3*OUT (erro nas entradas)

GEN

O gerador de função gera funções periódicas típicas.

As entradas são uma composição constituída do modo do tipo de contagem predefinido GEN_MODE, da BASE do tipo BOOL, do período (tipo TIME), de dois valores INT (CYCLES e AMPLITUDE) e da entrada RESET booleana.

O modo descreve a função que deve ser gerada..

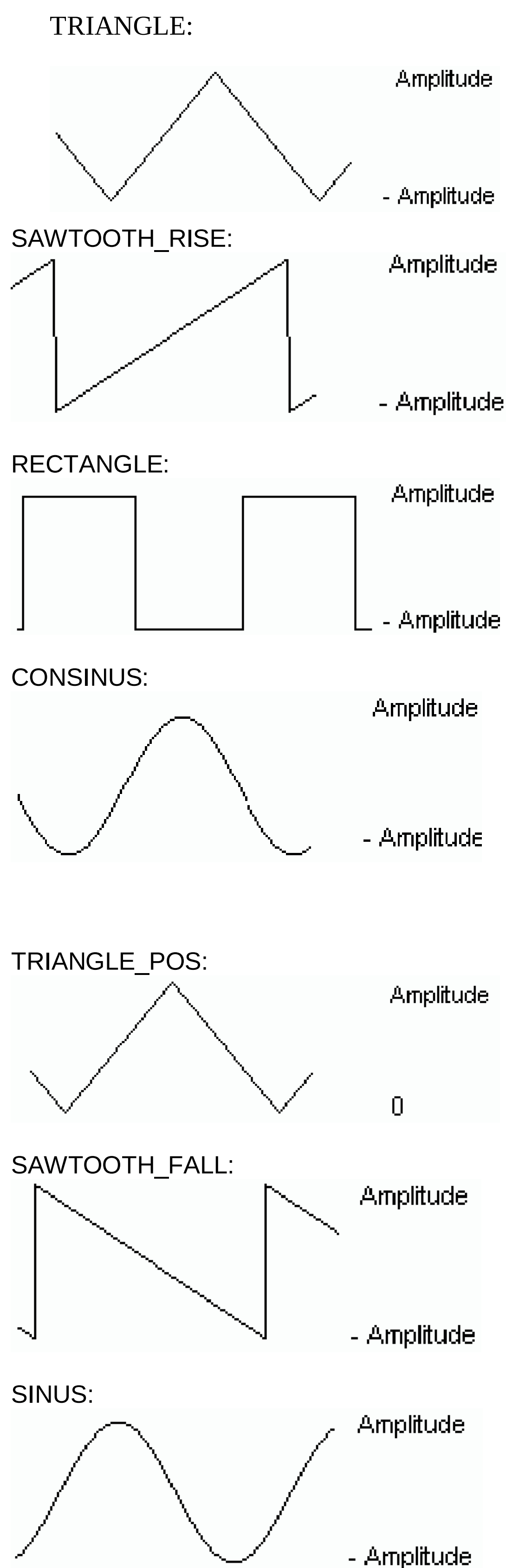


Figura 7-16. Caracterização da entrada GEN_MODE de GEN

BASE define se o período de ciclo está realmente relacionado com um tempo definido (BASE = TRUE) ou se ele está relacionado com um determinado número de ciclos, o que significa o número de chamadas do bloco funcional (BASE = FALSE).

PERIOD ou CYCLES define o período de ciclo correspondente.

AMPLITUDE define, de um modo trivial, a amplitude da função a ser gerada.

O gerador de função é mais uma vez definido como 0, logo que RESET = TRUE.

Exemplo em FBD:

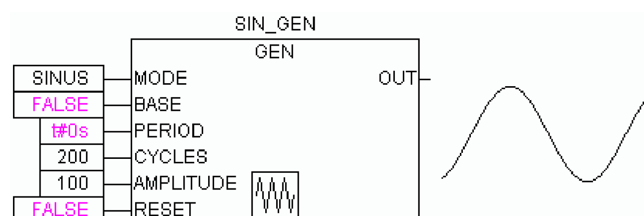


Figura 7-17. Bloco funcional GEN

Manipuladores de função

CHARCURVE

Este bloco funcional serve para representar valores, segmento por segmento, sobre uma função linear:



Figura 7-18. Bloco funcional CHARCURVE

IN do tipo INT é alimentado com o valor a ser manipulado. O BYTE N designa o número de pontos que define a função de apresentação. Esta linha característica, então, é gerada em um ARRAY P[0..10] com P do tipo POINT que é uma estrutura baseada em dois valores INT (X e Y).

A saída consiste em OUT, do tipo INT, o valor manipulado e o BYTE ERR, o qual irá indicar um erro, se necessário.

Os pontos P[0]..P[N-1] na ARRAY devem ser classificados de acordo com seus valores de X, caso contrário ERR recebe o valor 1. Se a entrada IN não está entre P[0].X e P[N-1].X, ERR=2 e OUT contém o correspondente valor limite P[0]. Y ou P[N-1].Y.

Se N está fora dos valores permitidos que estão entre 2 e 11, ERR, então, = 4.

Exemplo em ST:

Antes de mais nada, ARRAY P deve ser definida do cabeçalho:

```
VAR
...
 CHARACTERISTIC_LINE:CHARCURVE;
 KL:ARRAY[0..10] OF POINT:=(X:=0,Y:=0),(X:=250,Y:=50),
(X:=500,Y:=150),(X:=750,Y:=400),7((X:=1000,Y:=1000));
 COUNTER:INT;
 ...
END_VAR
```

Em seguida, fornecemos CHARCURVE com, por exemplo, um valor de incremento constante:

```
COUNTER:=COUNTER+10;
 CHARACTERISTIC_LINE(IN:=COUNTER,N:=5,P:=KL);
```

A figura a seguir ilustra o efeito:

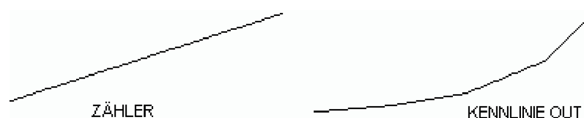


Figura 7-19. Traçado do efeito de CHARCURVE

RAMP_INT

RAMP_INT serve para limitar as rampas de subida ou descida da função de entrada. A entrada por um lado consiste de três valores INT: IN (entrada de função), ASCEND e DESCEND, o máximo acréscimo ou diminuição para um intervalo de tempo determinado, que é definido por TIMEBASE do tipo TIME. Definindo RESET para TRUE provoca inicialização de RAMP_INT.

A saída, do tipo INT contém o valor da função limitada (rampa de subida e descida).

Quando TIMEBASE estiver definido como t #0s, ASCEND e DESCEND não estão relacionadas com o intervalo de tempo, mas continuam as mesmas.

Exemplo em CFC:

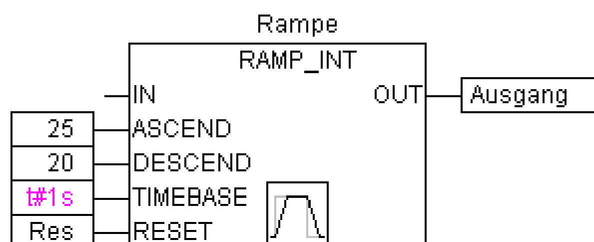


Figura 7-20. Exemplo em CFC de RAMP_INT

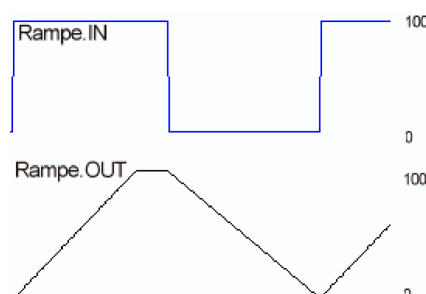


Figura 7-21. Entrada/saída de RAMP_INT

RAMP_REAL

RAMP_REAL funciona da mesma forma como RAMP_INT, com a diferença simples que as entradas IN, ASCEND, DESCEND e saída OUT são do tipo REAL.

Processamento de valores analógicos

HYSTERESIS

A entrada para este bloco funcional consiste de três valores INT: IN, HIGH e LOW. A saída OUT é do tipo BOOL.

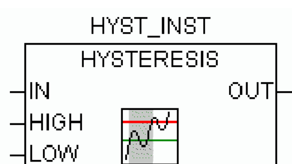


Figura 7-22. Bloco funcional Hysteresis

Se IN cai abaixo do valor limite LOW, OUT se torna TRUE. Se IN atingir limite superior HIGH, FALSE é fornecido.

Um exemplo ilustrativo:

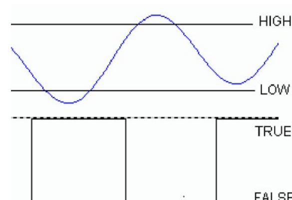


Figura 7-23. Exemplo do bloco funcional Hysteresis

LIMITALARM

Este bloco funcional especifica, se os o valor de entrada estão dentro de um intervalo definido e que limites violou quando isso aconteceu.

Os valores de entrada IN, HIGH e LOW são do tipo INT, enquanto as saídas O, U e IL são do tipo BOOL.

Se o limite superior HIGH é excedido por IN, O se torna TRUE e quando IN é inferior a LOW, U se torna TRUE. IL é TRUE se IN está compreendida entre LOW e HIGH.

Exemplo em FBD:

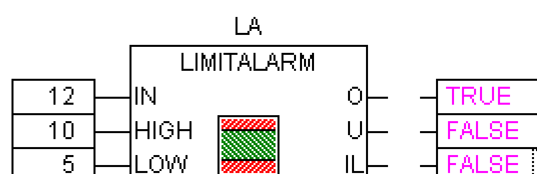


Figura 7-24. Bloco funcional LIMITALARM

Análise de expressões com a biblioteca AnalyzationNew.lib

Esta biblioteca fornece módulos para a análise de expressões. Se uma expressão composta é FALSE, os seus componentes que estão adicionados a este resultado podem ser avaliados. No Editor SFC o sinalizador SFCErrorAnalyzationTable usa esta função implicitamente para a análise de expressões em transições.

Exemplo de uma expressão:

B OR NOT (Y < X) OR NOT (NOT D AND E)

As funções:

As seguintes variáveis são usadas por todos os módulos:

InputExpr: BOOL, expressão a ser analisada

DoAnalyze: BOOL, TRUE inicia análise

ExpResult: BOOL, valor atual da expressão

Different é a saída do resultado da análise:

AnalyzeExpression retorna em uma string os componentes da expressão que estão adicionando ao valor total FALSE. A função AppendErrorString é usada com a finalidade de separar os componentes específicos na string de saída por caracteres " | ".

OutString: STRING, resultado da análise, seqüência dos componentes referentes à expressão (por exemplo, y < x | d)

AnalyseExpressionTable escreve os componentes da expressão adicionados ao valor total FALSE, para uma matriz. Para cada componente são fornecidas as seguintes informações pela estrutura ExpressionResult: nome, endereço, comentário, valor (atual).

OutTable: ARRAY [0..15] OF ExpressionResult.

Exemplo:

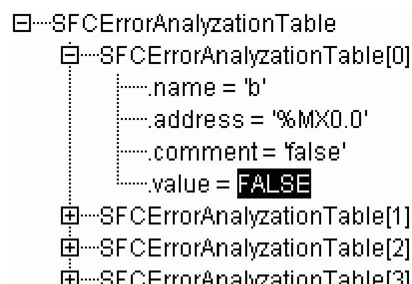


Figura 7-25. Exemplo de tabela "Analyse Expression"

AnalyseExpressionCombined combina as funcionalidades do AnalyzeExpression e do AnalyseExpressionTable.

Apêndice E: operadores e módulos de biblioteca

A tabela abaixo mostra uma visão geral sobre os operadores que estão disponíveis no MasterTool IEC e nas bibliotecas Standard.lib e Util.lib. Lá encontra-se a notação para ST e IL. Também para IL são listados os modificadores suportados.

Observe que para a coluna 'Operador IL': apenas a linha em que o operador é usado será exibida. Uma condição é que o (primeiro) operando necessário tenha sido carregado com êxito na linha anterior (por exemplo, em LD).

A coluna "Mod. IL' mostra os modificadores possíveis em IL:

C	O comando só é executado se o resultado da expressão anterior for TRUE.
N	Para JMPC, CALC, RETC: O comando só é executado se o resultado da expressão anterior for FALSE.
N	Caso contrário: negação do operando (e não do acumulador)
(	Operador delimitado por parênteses: somente depois de atingido o fechamento do parênteses é que a operação anterior aos mesmos será efetuada.

Tabela 7-18. Modificadores possíveis em IL

Por favor, obtenha uma descrição detalhada do uso a partir dos Anexos sobre operadores IEC adequados integrados no MasterTool IEC e nas bibliotecas.

Os operadores no MasterTool IEC:

Em ST	Em IL	Mod. IL	Descrição
'			Delimitadores de strings (por exemplo, ' string1')
.. **[ ]**			Tamanho do intervalo da Matriz (por exemplo, ARRAY[0..3] OF INT)
:			Delimitador entre Operando e Tipo em uma declaração (por exemplo, var1: INT;)
;			Terminação da instrução (por exemplo, a: = var1;)
^			Ponteiro de-referenciado (por exemplo, pointer1 ^)
	LD var1	N	Carregar o valor de var1 no buffer
:=	ST var1	N	Armazenar o resultado atual em var1
	S boolvar		Definir operando booleano boolvar exatamente para TRUE, quando o resultado atual é TRUE
	R boolvar		Definir operando booleano boolvar exatamente depois para FALSE, quando o resultado atual é TRUE
	JMP label	CN	Saltar para o rótulo
<Program name>	CAL prog1	CN	Chamar o programa prog1
<Instance name>	CAL inst1	CN	Chamar instância do Bloco funcional inst1
<Fctname>(vx, vy,..)	<Fctname> vx, vy	CN	Chamar a função fctname e transmitir as variáveis vx, vy
RETURN	RET	CN	Deixar a POU e voltar ao chamador
	(		O valor seguinte ao parêntese é tratado como operando, a operação antes do parêntese não é executada antes da expressão entre parênteses.
	)		Agora executar a operação que tenha sido definida anteriormente
AND	AND	N,(	E binário
OR	OR	N,(	OU binário
XOR	XOR	N,(	OU EXCLUSIVO binário
NOT	NOT		NÃO binário
+	ADD	(	Adição
-	SUB	(	Subtração

*	MUL	(	Multiplicação
/	DIV	(	Divisão
>	GT	(	Maior que
>=	GE	(	Maior ou igual
=	EQ	(	Igual
<>	NE	(	Diferente
<=	LE	(	Menor ou igual
<	LT	(	Menor que
MOD(in)	MOD		Módulo Divisão
INDEXOF(in)	INDEXOF		Índice interno de uma POU in1; [INT]
SIZEOF(in)	SIZEOF		Número de bytes solicitados para o tipo de dados (in)
SHL(K,in)	SHL		Deslocamento binário à esquerda do operando in por K
SHR(K,in)	SHR		Deslocamento binário à direita do operando in por K
ROL(K,in)	ROL		Rotação binária à esquerda do operando in por K
ROR(K,in)	ROR		Rotação binária à direita do operando in por K
SEL(G,in0,in1)	SEL		Seleção binária entre 2 operandos in0 (G é FALSE) e in1 (G é TRUE)
MAX(in0,in1)	MAX		Retorna o maior dos dois valores
MIN(in0,in1)	MIN		Retorna o menor dos dois valores
LIMIT(MIN,in,Max)	LIMIT		Limita o valor do intervalo (in é definido de volta para MIN ou MAX no caso de exceder o intervalo.
MUX(K,in0,...in_n)	MUX		Seleciona o K-ésimo valor de um grupo de valores (in0 a In_n)
ADR(in)	ADR		Endereço do operando em [DWORD]
ADRINST()	ADRINST()		Endereço da instância do Bloco funcional partir do qual se está chamando esse operador.
BITADR(in)	BITADR		Bitoffset do operando em [DWORD]
BOOL_TO_<type>(in)	BOOL_TO_<type>		Conversão de tipo do operando booleano
<type>_TO_BOOL(in)	<type>_TO_BOOL		Conversão de tipo para BOOL
INT_TO_<type>(in)	INT_TO_<type>		Conversão de tipo de um operando INT para outro tipo elementar
REAL_TO_<type>(in)	REAL_TO_<type>		Conversão de tipo de um operando REAL para outro tipo elementar
LREAL_TO_<type>(in)	LREAL_TO_<type>		Conversão de tipo de um operando LREAL para outro tipo elementar
TIME_TO_<type>(in)	TIME_TO_<type>		Conversão de tipo de um operando de TEMPO para outro tipo elementar
TOD_TO_<type>(in)	TOD_TO__<type>		Conversão de tipo de um operando TOD para outro tipo elementar
DATE_TO_<type>(in)	DATE_TO_<type>		Conversão de tipo de um operando de DATA para outro tipo elementar
DT_TO_<type>(in)	DT_TO_<type>		Conversão de tipo de um operando DT para outro tipo elementar
STRING_TO_<type>(in)	STRING_TO_<type>		Conversão de tipo de um operando string para outro tipo elementar, in deve conter o valor válido do tipo desejado
TRUNC(in)	TRUNC		Conversão de REAL para INT
ABS(in)	ABS		Valor absoluto do operando in
SQRT(in)	SQRT		Raiz quadrada do operando in
LN(in)	LN		Logaritmo natural do operando in
LOG(in)	LOG		Logaritmo do operando in, base 10
EXP(in)	EXP		Função exponencial do operando in
SIN(in)	SIN		Seno de operando in
COS(in)	COS		Cosseno do operando in

TAN(in)	TAN		Tangente de operando in
ASIN(in)	ASIN		Arco seno do operando in
ACOS(in)	ACOS		Arco cosseno do operando in
ATAN(in)	ATAN		Arco tangente do operando in
EXPT(in,expt)	EXPT expt		Exponenciação do operando com expt

Tabela 7-19. Os operadores no MasterTool IEC

Elementos da Standard.lib:

Em ST	**Em IL**	**Descrição**
LEN(in)	LEN	Comprimento da string do operando in
LEFT(str,size)	LEFT	STRING inicial esquerda de um dado tamanho da string str fornecida
RIGHT(str,size)	RIGHT	STRING inicial direita de um dado tamanho da string str fornecida
MID(str,size,pos)	MID	STRING parcial de str de dado tamanho na posição pos
CONCAT('str1','str2')	CONCAT 'str2'	Concatenação de duas STRINGs subseqüentes
INSERT('str1','str2',pos)	INSERT 'str2',p	Inserir seq1 na string str2 na posição pos
DELETE('str1',len,pos)	DELETE len,pos	Excluir STRING parcial (comprimento tam), iniciar na posição pos de str1
REPLACE('str1','str2',len,pos)	REPLACE 'str2',len,pos	Substituir a cadeia parcial de tam. lenght por str2, iniciar na posição pos de str1
FIND('str1','str2')	FIND 'str2'	Procurar pela STRING parcial str2 em str1
SR	SR	Bistable FB (SET dominante)
RS	RS	Bistable FB (RESET dominante)
SEMA	SEMA	FB: Software Semáforo (interrupção)
R_TRIG	R_TRIG	FB: detecção de borda de subida
F_TRIG	F_TRIG	FB: detecção de borda de descida
CTU	CTU	FB: incrementar
CTD	CTD	FB: decrementar
CTUD	CTUD	FB: incrementar e decrementar
TP	TP	FB: dispara
TON	TON	FB: temporizador na energização
TOF	TOF	FB: temporizador na desenergização
RTC	RTC	FB: relógio de tempo real

Tabela 7-20. Elementos da Standard.lib

Elementos da Util.lib:

BCD_TO_INT	Conversão: BCD para INT
INT_TO_BCD	Conversão (Byte): INT para BCD
EXTRACT(in,n)	O n-ésimo bit de DWORD in retorna em BOOL
PACK	8 bits para byte
PUTBIT	Um bit de uma DWORD é definido para um certo valor
UNPACK	Um Byte retorna como bits únicos
DERIVATIVE	Derivação local
INTEGRAL	Integral
LIN_TRAFO	Transformação de valores reais
STATISTICS_INT	Min.,Max, Média (INT)
STATISTICS_REAL	Min.,Max, Média (REAL)
VARIANCE	Variância
PD	Controlador PD
PID	Controlador PID
BLINK	Sinal pulsante
FREQ_MEASURE	Freqüência medida de sinal de entrada booleano
GEN	Funções periódicas
CHARCURVE	Funções lineares
RAMP_INT	Limitação (rampa) de função (INT)
RAMP_REAL	Limitação (rampa) de função (REAL)
HYSTERESIS	Histerese
LIMITALARM	Monitora se valor de entrada excede os limites de um intervalo definido

Tabela 7-21. Elementos da Util.lib

Apêndice F: linha e arquivo de comando

Comandos de Linha de Comando

Quando o MasterTool IEC é iniciado, pode-se adicionar comandos na linha de comando durante a execução do programa. Esses comandos começam com um "/". Não é considerado o uso de letras maiúsculas/minúsculas. Os comandos serão executados seqüencialmente da esquerda para a direita.

/online	Imediatamente após o início o MasterTool IEC tenta ficar online com o projeto atual.
/run	Após o início da sessão o MasterTool IEC inicia o programa aplicativo. Só é válido em combinação com /online.
/batch	O MasterTool IEC inicia sem interface do usuário e retorna o código de erro do primeiro erro e o valor do retorno do primeiro comando que é encerrado com uma advertência. O MasterTool IEC encerrará imediatamente após o arquivo de comando ter sido processado. O processamento do arquivo de comando será anulado logo que o primeiro comando é processado com um erro. Advertências não encerram o processamento. Se não ocorrerem erros nem advertências, o valor de retorno é S_OK.
/show ... **/show hide** **/show icon** **/show max** **/show normal**	Configurações para a janela do quadro MasterTool IEC podem ser feitas. A janela não será exibida, ela também não será representada no menu de tarefas. A janela será minimizada na exibição. A janela será maximizada na exibição. A janela será exibida no mesmo status, como era durante o último fechamento.
/out <outfile>	Todas as mensagens são exibidas na janela de mensagens e além disso, são gravadas no arquivo <outfile>.
/noinfo	Sem tela inicial no início do MasterTool IEC
/userlevel <group>	Definição do grupo de usuários (por exemplo, " / userlevel 0 " para o usuário grupo 0)
/password <password>	Entrada direta da senha do grupo de usuário (por exemplo, " / senha abc ")
/openfromplc	O projeto que está disponível no momento no sistema do dispositivo ligado, será carregado.
/visudownload	Se MasterTool IEC HMI for iniciado com um projeto, que não coincide com o projeto atualmente disponível no sistema do dispositivo, um download será oferecido. (Diálogo, fechado com SIM ou NÃO).
/notargetchange	Uma alteração no sistema do dispositivo só pode ser feita através de um arquivo de comando. Consultar o comando "dispositivo...".
/cmd <cmdfile>	Depois de iniciar os comandos, <cmdfile> é executado.

Tabela 7-22. Comandos de Linha de Comando

Considere a seguinte sintaxe para uma linha de comando:

"< Caminho do MasterTool IEC-exe-arquivo >" "< Caminho do projeto >" /<command1> /<command2>....

Exemplo de uma linha de comando:

O projeto ampel.pro é aberto, mas nenhuma janela é aberta. Os comandos incluídos no arquivo de comando command.cmd serão executados.

"D:\dir1\MasterTool IEC" "C:\projects\ampel.pro" /show hide /cmd command.cmd

Comandos Comando de arquivo (cmdfile)

Consultar a tabela seguinte para obter uma lista de comandos que podem ser usados em um arquivo de comando (<cmdfile>). Pode-se então chamar o arquivo de comando por uma linha de comando

(ver acima). Não há sensitividade de maiúsculas e minúsculas. A linha de comando será exibida como uma mensagem na janela de mensagens e pode estar disponível em um arquivo de mensagem (ver abaixo) exceto se o comando é antecedido por um "@".

Todos os sinais após um ponto-e-vírgula (;) serão ignorados (comentário). Parâmetros contendo espaços em branco devem estar entre aspas. *Umlauts* só podem ser usados se o arquivo de comando é criado no código ANSI. Palavras-chave podem ser usadas nos parâmetros de comando. Uma lista de palavras-chave está localizada após as descrições de comando a seguir.

Comandos para controlar os comandos subsequentes:

onerror continue

Os comandos subseqüentes serão executados mesmo se ocorrer um erro.

onerror break

Os comandos subseqüentes não serão mais executados se tiver sido detectado um erro.

Comandos do menu online:

online login

Login com o projeto carregado (‘Comunicação’ 'Login')

online logout

Logout (‘Comunicação’ 'Logout')

online run

Iniciar programa aplicativo (‘Comunicação’ 'Run')

online stop

Parar o programa aplicativo (‘Comunicação’ 'Stop')

online bootproject

Criação de um projeto de inicialização. Este comando pode ser aplicado nos modos offline e online! (Veja também descrição do comando ‘Comunicação’ 'Criar projeto de inicialização'!)

online sourcecodedownload

Download do código de fonte do projeto para o CP (‘Comunicação’ ‘Download do código fonte’)

online sim

Alternar para modo de simulação (‘Comunicação’ ‘Modo simulação’)

online sim off

Sair do modo de simulação (‘Comunicação’ ‘Modo simulação’)

Comandos do menu do arquivo:

file new

Um novo projeto é criado ('Arquivo' 'Novo')

file open <projectfile>

Possíveis acréscimos:

O projeto <projectfile> será carregado ('Arquivo' 'Abrir')

/readpwd:<readpassword>

A senha para acesso de leitura é dada aqui, então não aparecerá diálogo para pedir a senha quando o projeto protegido contra leitura é aberto.

/writepwd:<writepassword>

A senha de acesso completo é fornecida aqui, então não aparecerá diálogo para pedir a senha quando o projeto é aberto.

file close

O projeto atual será fechado ('Arquivo' 'Fechar')

file save

O projeto atual será salvo ('Arquivo' 'Salvar')

file saveas <projectfile>

Acréscimo opcional: <type><version>

O projeto atual será salvo com o nome de arquivo <projectfile> ('Arquivo' 'Salvar como')

Padrão: o projeto será salvo como <projectfile>.pro sob a versão atual do MasterTool IEC. Para salvar o projeto como uma biblioteca interna ou externa ou como projeto com uma versão mais antiga do MasterTool IEC, adicione o respectivo comando:

Entradas possíveis para <type>:

"internallib" Salva como biblioteca interna:
"externallib" Salva como biblioteca externa
"pro" Salva como projeto em uma versão mais antiga

entradas válidas para <Version>: 15, 20, 21, 22 (versões de produto 1.5, 2.0, 2.1, 2.2)

Exemplo: "file save as lib_xy internallib22" -> O projeto "projeto XY.pro", que é criado na atual versão do MasterTool IEC será salvo como" lib_xy.lib" para V2.2.

file saveas <projectfile>

O projeto atual sera salvo com o nome de arquivo <projectfile> ('Arquivo' 'Salvar como')

file printersetup <filename>.dfr

optionally add: pageperobject or

pagepersubject

Defina um arquivo de documentação ('Arquivo' 'Configurações de impressão') e opcionalmente defina uma das opções de impressão 'Nova página por objeto' ou 'Nova página por sub-objeto' ; estes ajustes afetam a impressão do documento (documentação de projeto, veja abaixo)

file archive <filename>.zip

O projeto será arquivado em um arquivo ZIP com o nome dado ('Arquivo' 'Salvar/enviar arquivo')

file quit

MasterTool IEC será fechado ('Arquivo' 'Sair')

Lista de comandos do menu de projeto:

project build

= ('Projeto' 'Compilar')

project rebuild ou project compile

= ('Projeto' 'Compilar tudo')

project clean

= ('Projeto' 'Limpar projeto')

project check

= ('Projeto' 'Verificar tudo')

project compile

= ('Projeto' 'Compilar tudo')

project check

= ('Projeto' 'Verificar')

project build

= ('Projeto' 'Compilar')

project import <file1> ... <fileN>

= ('Projeto' 'Importar') os arquivos <file1> ... <fileN> Considere: seleções podem ser usadas, p. e., "Importar projeto C:\projects*.exp" importará todos os arquivos com a extensão *.exp encontrados no diretório C:\projects.

project export <expfile>

= ('Projeto' 'Exportar') <arqexp>

project expmul

Cada objeto do projeto atual será exportado em um arquivo próprio, o qual recebe o nome do objeto

project documentation

O projeto inteiro será impresso na impressora padrão. ('Projeto' 'Documentação', veja também acima "file printersetup")

Comandos para o controle do arquivo de mensagens:
out open <arqmens>

O arquivo <arqmens> abre como arquivo de mensagem. Novas mensagens serão anexadas.

out close

O arquivo de mensagem mostrado atualmente será fechado

out clear

Todas as mensagens do arquivo de mensagem mostrado atualmente serão excluídos.

Comandos para o controle de mensagens:
echo on

As linhas de comando serão mostradas como mensagens.

echo off

As linhas de comando não serão mostradas como mensagens.

echo <texto>

<texto> será mostrado na janela de mensagens

Comandos para o controle da substituição de objetos respectivamente para o controle de arquivos para importação, exportação e cópia:
replace yesall

Substituir tudo (qualquer comando 'query on' será ignorado; diálogos não abrirão).

replace noall

Substituir nenhum (qualquer comando 'query on' será ignorado; diálogos não abrirão).

replace query

Se um comando 'query on' é definido, então um diálogo abrirá referente à substituição dos objetos mesmo se houver um comando 'replace yesall' ou 'replace noall'.

Comandos para o controle dos parâmetros default dos diálogos do MasterTool IEC:
query on

Diálogos são exibidos e necessita entrada do usuário.

query off ok

Todos os diálogos respondem como se o usuário tivesse pressionado o botão 'OK'.

query off no

Todos os diálogos respondem como se o usuário tivesse pressionado o botão 'Não'.

query off cancel

Todos os diálogos respondem como se o usuário tivesse pressionado o botão 'Cancelar'.

Comandos para arquivos de comando de chamadas como sub-programas:

call <parâmetro1> ... <parâmetro10>

Arquivos de comando são chamados como sub-programas. Até 10 parâmetros podem ser passados. No arquivo que é chamado, os parâmetros podem ser acessados com $0 - $9.

call <parâmetro1> ... <parâmetro10>

Arquivos de comando são chamados como sub-rotinas. Até 10 parâmetros podem ser consignados. Na sub-rotina chamada, os parâmetros podem ser acessados com $0 - $9.

Configuração dos diretórios usados pelo MasterTool IEC: (-> diálogo de opções de projeto, categoria 'Diretórios', sub-categoria 'Geral'):

Se vários diretórios são definidos com um dos seguintes comandos, estes devem ser separados por um ponto e vírgula + espaço em branco e a linha inteira dos diretórios deve estar entre aspas

Exemplo, dois caminhos: `dir lib "D:\MasterTool IEC\Libraries\Standard; D:\MasterTool IEC\Libraries\NetVar"`

dir lib <libdir>

Define <libdir> como o diretório da biblioteca.

dir compile <compiledir>

Define <compiledir> como o diretório para os arquivos de compilação.

dir config <configdir>

Define <configdir> como o diretório para os arquivos de configuração.

dir upload <uploaddir>

Define <uploaddir> como o diretório para os arquivos de upload.

Atraso no processamento do CMDFILE:

delay 5000

Espera 5 segundos

Controlando o Gerenciador de Monitoração e Receitas:

watchlist load <arquivo>

= ('Extras' 'Carregar lista de monitoração').

watchlist save <arquivo>

= ('Extras' 'Salvar lista de monitoração').

watchlist set <texto>

A lista de monitoração é ativada (corresponde a selecionar uma lista de monitoração na parte esquerda do gerenciador específico).

watchlist read

= ('Extras' 'Ler receita').

watchlist write

= ('Extras' 'Escrever receita').

Vinculando bibliotecas:

library add <bib. arq1> <bib. arq2> .. <bib. arqN>

Anexa o arquivo da biblioteca especificada à lista de bibliotecas do projeto atualmente aberto. Se o caminho do arquivo é relativo, o diretório da biblioteca incluída no projeto é usado como a raiz do caminho.

library delete [<bib.1> <bib.2> .. <bib.N>]

Apaga as bibliotecas especificadas da lista de bibliotecas do projeto atualmente aberto.

Copiando objetos:

object copy <arqprjfonte> <caminhofonte> <caminhodestino>

Copia objetos do caminho especificado do arquivo de projeto fonte para o caminho do dispositivo do projeto já aberto. Se o caminho da fonte é o nome de um objeto, este será copiado. Neste caso, a estrutura da pasta abaixo da pasta da fonte será duplicada. Se o caminho do dispositivo não existe ainda, ele será criado.

Acesso somente leitura para objetos particulares:

object setreadonly <TRUE|FALSE> <objtipo> | <nomeobj>

Define acesso somente leitura para um objeto. Define o tipo de objeto e em caso de tipos pou, dut, gvl, vis também o nome do objeto.

Tipos de objetos possíveis: pou, dut (tipos de dados), gvl (lista de variáveis globais), vis (visualização), cnc (CNC), liblist (bibliotecas), targetsettings (configurações do dispositivo), toolinstanceobject (instâncias de ferramentas particulares), toolmanagerobject (todas as instâncias na árvore de ferramentas), customplconfig (configuração do CP), projectinfo (informação do projeto), taskconfig (configuração de tarefas), trace, watchentrylist (Gerenciador de Monitoração e Receitas), alarmconfig (configuração de alarmes).

P. e., "object setreadonly TRUE pou plc_prg" definirá PLC_PRG para acesso somente leitura

Entrando parâmetros de comunicação (gateway, dispositivo):

gateway local

Define o gateway no computador local como o gateway atual.

gateway tcpip <End> <Porta>

Define o gateway no computador remoto especificado como o gateway atual:

- <End>: endereço TCP/IP ou hostname do computador remoto
- <Porta>: porta TCP/IP do gateway remoto

Importante: somente gateways que não têm definição de senha podem ser selecionados!

device guid <guid>

Define o dispositivo com o GUID especificado como o dispositivo em uso.

GUID deve ter o seguinte formato:

{01234567-0123-0123-0123-0123456789ABC}

As chaves e os hífens devem aparecer nas posições especificadas.

device instance <nomeinst>

Define o nome da instância do dispositivo atual para o nome especificado.

device parameter <Id> <valor>

Atribui o valor especificado, o qual será então interpretado pelo dispositivo, para o parâmetro com a ID especificada.

Chamadas de sistema:

system <comando>

Executa o comando do sistema operacional especificado.

Seleciona sistema do dispositivo:

target <Id>

Define a Plataforma do Dispositivo do projeto atual. Se o MasterTool IEC é iniciado com a opção de linha de comando "/notargetchange", somente por este comando um dispositivo pode ser definido.

Query estado do sistema:

state offline

Retorna "S_OK", se atualmente não há conexão entre o sistema de programação e o sistema do dispositivo (modo offline), caso contrário, "HRESULT[0x800441f0)" (modo online).

state online

Retorna "S_OK", se atualmente há conexão entre o sistema de programação e o sistema do dispositivo (modo online), caso contrário, "HRESULT[0x800441f0)" (modo offline).

Senha para grupo de usuários:

Com os seguintes ajustes, quando da abertura de um projeto que está protegido por senha de grupo de usuários, pode-se entrar com a senha para certo grupo de usuário. Entretanto, mesmo se a exibição do diálogo de entrada do usuário é aberta via arquivo de commando. As entradas para o grupo de usuário e a senha devem ser colocados antes do comando “arquivo abrir…”!

Exemplo:

user level 0

user password aaa

file open "D:\MasterTool IEC\projects\xxxx.pro"

query off ok

user level

Grupo de usuário (senha definida no projeto obtida pelo comando subseqüente “user password”).

user password

Senha para grupo de usuário especificada pelo comando precedente "user group".

Configurações de visualização:

visual settings...

= 'Extras' 'Configurações', categoria Idioma (ou Configurações do dispositivo, categoria Visualização).

... language file on || off

Opção 'Arquivo de idioma’ ativado (on) ou desativado (off). No caso de ativação, a opção ‘Textos dinâmicos’ será desativada.

... set languagefile <path language file>

Especifica o arquivo de idioma a ser usado (.tlt or .vis).

Exemplo: "visual settings set languagefile proj1.tlt"

... dynamictexts on || off

Opção ‘Textos dinâmicos’ ativado (on) ou desativado (off). No caso de ativação, a opção ‘Arquivo de idioma’ será desativada.

... dynamictextfiles <file path> | <file path> | ...

Especificação de uma lista de caminhos de arquivos de idioma a ser usada. A lista anterior será apagada.

Exemplo: "visual settings D:\dynfiles\p1.xml D:\dynfiles\p2.xml"

... dynamictexthideelements on || off

Ativar/desativar opção ‘Suprimir elementos se nenhum texto de substituição…’

... language <language>

Especifica o idioma a ser usado.

Exemplo: "visual settings language German"

... tablekeyboardusage_web on || off

Ativar/desativar opção ‘Funções do teclado para tabelas’ na Visualização Web (Configurações do dispositivo).

... tablekeyboardusage_mastertooliec on || off

Ativar/desativar opção ‘Funções do teclado para tabelas’ no MasterTool IEC e MasterTool IEC HMI (Configurações do dispositivo)

visual webvisuactivation on || off

Ativar/desativar opção Visualização Web (Configurações do dispositivo) .

Comandos relacionados ao gerenciamento do projeto na base de dados ENI:

Na descrição dos comandos, espaços reservados são usados:

- <categoria>: Substituir por "project" ou "shared" ou "compile" dependendo da categoria da base de dados
- <NomePOU>: nome do objeto
- <Objtipo>: Substituir pelo atalho, o qual é anexado como uma extensão ao nome da POU do objeto na base de dados e que reflete o tipo do objeto. Exemplo: Objeto "GLOBAL_1.GVL" -> o nome da POU é "GLOBAL_1", o tipo do objeto é "GVL" (global variables list)
- <comment>: Substituir por um texto de comentário (entre aspas simples), o qual será armazenado no histórico da versão com a ação particular.

eni on
eni off

Ativar/desativar opção 'Usar controle de fonte (ENI)' (Diálogo 'Projeto' 'Opções' 'Controle de fonte de projeto').

eni project readonly on
eni project readonly off

Ativar/desativar opção 'Somente leitura' para a categoria base de dados 'Objetos do projeto' (Diálogo 'Projeto' 'Opções' 'Objetos do projeto').

eni shared readonly on
eni shared readonly off

Ativar/desativar opção 'Somente leitura' para a categoria ‘Objetos compartilhados’ (Diálogo 'Projeto' 'Opções' 'Objetos compartilhados').

eni set local <NomePOU>

O objeto será atribuído à categoria ‘Local’, isto é, não será armazenado na base de dados do projeto (Diálogo 'Projeto' 'Objeto' 'Propriedades' 'Conexão da base de dados').

eni set shared <NomePOU>

O objeto será atribuído à categoria 'Objetos compartilhados', isto é, não será armazenado na base de dados do projeto (Diálogo 'Projeto' 'Objeto' 'Propriedades' 'Conexão da base de dados').

eni set project <NomePOU>

O objeto será atribuído à categoria 'Objetos do projeto' (Diálogo 'Projeto' 'Objeto' 'Propriedades' 'Conexão da base de dados').

eni <categoria> server <TCP/IP_End> <Porta> <Nomeprj> <Nomeusr> <Senha>

Configura a conexão para o servidor ENI na categoria 'Objetos do projeto' (Diálogo 'Projeto' 'Opções' 'Base de dados do projeto'.

Exemplo:
eni project server localhost 80 batchtest\project EniBatch Batch

(TCP/IP_End = localhost, Porta = 80, Nomeprj = batchtest\project, Nomeusr = EniBatch, Senha = Batch)

eni compile sym on
eni compile sym off

Ativar/desativar opção 'Criar informações de símbolos ASCII' para os objetos da categoria 'Compilar arquivos' (Diálogo 'Projeto' 'Opções' 'Controle de fonte de projeto'...).

eni compile sdb on
eni compile sdb off

Ativar/desativar opção 'Criar informações de símbolos binários' para os objetos da categoria 'Compilar arquivos' (Diálogo 'Projeto' 'Opções' 'Controle de fonte de projeto'...).

eni compile prg on
eni compile prg off

Ativar/desativar opção 'Criar projeto de inicialização' para os objetos da categoria 'Compilar arquivos' (Diálogo 'Projeto' 'Opções' 'Controle de fonte de projeto'...).

Comandos do menu 'Projeto' 'Base de dados do projeto' para trabalhar com a base de dados:
eni set <categoria>

O objeto é atribuído à categoria da base de dados nomeada ('Definir')

'eni set <categoria>set <Objtipo>:<NomePOU> <Objtipo>:<NomePOU>

O objeto que está listado separado por espaços será atribuído à categoria da base de dados nomeada ('Múltiplas definições') .

Exemplo:
"eni set project pou:as_fub pou:st_prg"

-> os objetos (POU) as_fub e st_prg são atribuídos à categoria 'Objetos do projeto'.

eni <categoria> getall

A última versão de todos os objetos da categoria nomeada ('Obter todas últimas versões').

'eni <categoria>get <Objtipo>:<NomePOU> <Objtipo>:<NomePOU>

Os objetos da categoria nomeada que estão listados separados por espaços serão chamados a partir da base de dados ('Múltiplas definições') ('Obter todas últimas versões').

Exemplo:
"eni project get pou:as_fub gvl:global_1"

-> a POU as_fub.pou e as variáveis globais list global_1.gvl serão chamadas a partir da base de dados.

eni <categoria> checkoutall "<comment>"

Todos os objetos da categoria nomeada serão submetidos a check out a partir da base de dados. Os comentários definidos serão armazenados com a ação de check out no histórico da versão.

eni <categoria> checkout "<comment>" <Objtipo>:<NomePOU> <Objtipo>:<NomePOU>

Todos os objetos (objtipo: nomePOU) que estão listados separados por espaços serão submetidos à check out da base de dados. Os comentários definidos serão armazenados com a ação de check out no histórico da versão para cada objeto em particular.

Exemplo:
"eni project checkout " for working on xy" pou:as_fub gvl:global_1"

-> A POU as_fub e as variáveis globais list global_1 serão submetidos a check out e o comentário "for working on xy" será armazenado com esta ação.

eni <categoria>checkinall "<comment>"

Todos objetos do projeto, que estão sob controle de fonte na base de dados do projeto, serão submetidos a check in. Os comentários definidos serão armazenados com a ação de check in.

eni <categoria> checkin "<comment>" <Objtipo>:<NomePOU> <Objtipo>:<NomePOU>

Todos os objetos (objtipo: nomePOU) que estão listados separados por espaços serão submetidos à check in para a base de dados. Os comentários definidos serão armazenados com a ação de check in no histórico da versão para cada objeto em particular (veja acima: check out).

Palavras-chave para os parâmetros de comando:

As seguintes palavras-chave, entre “$”, podem ser usadas em parâmetros de comando:

$PROJECT_NAME$

Nome do projeto atual (sem extensão ".pro").

$PROJECT_PATH$

Caminho do diretório onde o projeto atual está (sem indicação de drive e sem barra invertida no final, p. e. "projects\sub1").

$PROJECT_DRIVE$

Drive onde o projeto atual está (sem barra invertida no final, p. e. "D:").

$COMPILE_DIR$

Diretório de compilação do projeto atual (sem indicação de drive e sem barra invertida no final, p. e. "D:\mastertooliec\compile").

EXE_DIR

Diretório onde o arquivo mastertooliec.exe está (sem indicação de drive e sem barra invertida no final, p. e. D:\mastertooliec).

Exemplo de um arquivo de comando:

Um arquivo de comando como mostrado abaixo abrirá o arquivo de projeto ampel.pro, então carregará uma lista de monitoração, a qual estava armazenada como w.wtc, então iniciará o programa aplicativo e escreverá, após uma espera de 1 segundo, os valores das variáveis na lista de monitoração watch.wtc (que sera salva) e finalmente fechará o projeto.

```
file open C:\projects\MasterTool IEC_test\ampel.pro
query off ok
watchlist load c:\work\w.wtc
online login
online run
delay 1000
```

```
watchlist read
watchlist save $PROJECT_DRIVE$\$PROJECT_PATH$\w_update.wtc
online logout
file close
```

Este arquivo de comando abrirá o projeto ampel.pro, carregará uma lista de monitoração existente w.wtc, iniciará o programa aplicativo e escreverá, após uma espera de 1 segundo, os valores das variáveis na lista de monitoração w_update.wtc, a qual sera salva no diretório C:\projects\MasterTool IEC_test" e então fechará o projeto novamente.

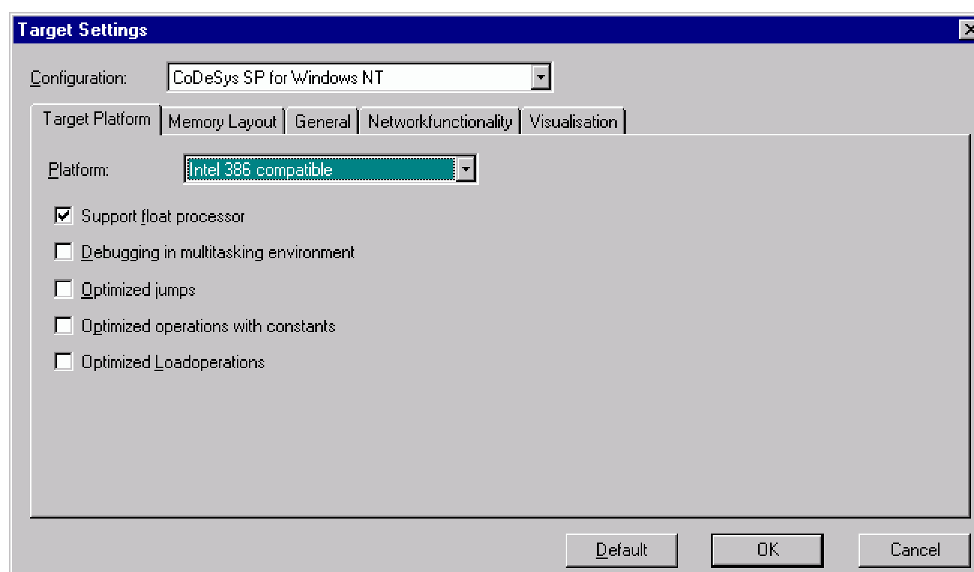
Um arquivo de comando é chamado em uma linha de comando como mostrado aqui:

"<path of MasterTool IEC.exe>" /cmd "<path of cmd file>"

Apêndice G: configurações do dispositivo em detalhes

Sistema do dispositivo 'Intel 386 compatível', Plataforma do Dispositivo

Diálogo Configurações do dispositivo 'Intel 386 compatível', Plataforma do Dispositivo



7-26. Diálogo Configurações do dispositivo 'Intel 386 compatível', Plataforma do Dispositivo

Item do diálogo	Significado
Plataforma	Tipo do sistema do dispositivo
Suportar processador ponto flutuante	Se ativado: comandos FPU são gerados para operações de ponto flutuante
Depuração em ambiente multi-tarefas	Se ativado:código adicional é gerado, o que permite operações de depuração em ambientes multi-tarefas
Saltos otimizados	Se ativado: saltos condicionais otimizados após operações de comparação; mais rápido +menos código (especialmente em 386/486); linhas contendo condições antes dos saltos serão exibidas na cor cinza no modo de controle de fluxo
Operações otimizadas com constantes	Operações otimizadas com constantes (A = A +1, A < 500, etc.); mais rápido + menos código (especialmente em 386/486); constantes serão monitoradas na cor cinza no modo de controle de fluxo
Operações de carga otimizadas	Operações de carga não serão executadas em acesso múltiplos a uma variável/constante; mais rápido + menos código

Tabela 7-23. Diálogo Configurações do dispositivo 'Intel 386 compatível', Plataforma do Dispositivo

Sistema do dispositivo Motorola 68K, Categoria Plataforma do Dispositivo

Diálogo Configurações do dispositivo 'Motorola 68K', Plataforma do Dispositivo

Item do diálogo	Significado
Plataforma	Tipo do dispositivo
UCP	Variante da UCP 68k: versão básica 6800 ou UCP32 e superiores
Suportar processador ponto flutuante	Se ativado: comandos FPU são gerados para operações de ponto flutuante
Usar offset de salto de 16 bits	Se ativado: saltos para avaliar expressões booleanas trabalham com offsets relativos de 16 bits (expressões mais complexas são possíveis, mas com código maior) Se não ativado: offsets de 8 bits são usados
Permitir estruturas alinhadas de bytes	Se ativado: apenas endereçamento par Se não ativado: endereçamento de endereços ímpares também possíveis
Registrador	A2, A4, A5, A6: o registrador de endereço indicado está reservado e não é

reservado 1	usado. Se nenhum: ele pode ser usado pelo gerador de código
Registrador reservado 2	Registrador de endereço reservado adicional. O registrador de endereço indicado é reservado e não usado. Se "Nenhum" pode ser usado pelo gerador de código
Registro-base para dados de biblioteca	Registro para endereçamento estático de dados dentro de bibliotecas C (antes de chamar as funções de biblioteca ele é carregado com o endereço de memória livre). Se "Nenhum" A5 é usado como valor pré-definido
Modo de saída	Nada = nenhuma saída Assembler = durante a compilação um arquivo "code68k.hex" é criado no diretório de compilação (configuração em "Projeto/Opções/Diretórios"). Ele contém o código assembler gerado. Disassembler = em acréscimo ao 1, o arquivo contém o código Disassembler.

Tabela 7-24. Diálogo Configurações do dispositivo 'Motorola 68K', Plataforma do Dispositivo

Sistema do dispositivo Infineon C16x, Categoria Plataforma do Dispositivo

Diálogo Configurações do dispositivo 'Infineon C16x', Plataforma do Dispositivo

Item do diálogo	Significado
Plataforma	Tipo do dispositivo
Código / Compilador:	Compilador usado durante compilação do sistema do dispositivo e das bibliotecas (em conformidade com as convenções de chamadas em C)
Código / Tamanho da pilha	Máxima "profundidade" de chamada (encadeamento)
Código / Dados	Modelo de memória para dados
Código / Funções	Modelo de memória para código
Ini. funções	Se ativado: funções contêm código de inicialização para variáveis locais
Otimizar	Se ativado: otimização de código para índices de matrizes constantes
Saída ...HEX	Se ativado: saída de um Hex-Dump do código
Saída ...BIN	Se ativado: saída de um arquivo binário do código
Saída ...MAP	Se ativado: saída de um arquivo MAP do código
Saída ...LST	Se ativado: saída de um arquivo LIST do código
Saída LST, dos endereços	Se ativado: saída da lista dos endereços
DPPs / DPP0..DPP2 Em instâncias	Apontadores de página de dados são setados DPP para DPP0, DPP1, DPP2 DPP para endereçamento curto de instâncias de blocos funcionais
Bibliotecas / Código Tabelas Dados Comprimento de dados POUs Referências	Ajustes para bibliotecas. Endereços iniciais para código, tabelas, dados, comprimento de dados, blocos e referências

Tabela 7-25. Diálogo Configurações do dispositivo 'Infineon C16x', Plataforma do Dispositivo

Sistemas De dispositivo Intel StrongARM e Power PC, Categoria Plataforma do Dispositivo

Os itens de diálogo para esses dois sistemas do dispositivo são idênticos.

Diálogo Configurações do dispositivo 'PowerPC', Plataforma do Dispositivo

Item do diálogo	Significado
Plataforma	Tipo do dispositivo
Suportar processador ponto flutuante	Se ativado: comandos FPU são gerados para operações de ponto flutuante
Registrador do primeiro parâmetro (inteiro)	Registro onde o primeiro parâmetro inteiro das chamadas de funções C é transmitido (intervalo depende do sistema operacional)
Registrador do último parâmetro (inteiro)	Registro onde o último parâmetro inteiro das chamadas de funções C é transmitido (intervalo depende do sistema operacional)

Registrador para valores de retorno (inteiro)	Registro onde o parâmetro inteiro das chamadas de funções C é retornado (intervalo depende do sistema operacional)
Registrador do primeiro parâmetro (flutuante)	Registro onde o primeiro parâmetro flutuante das chamadas de funções C é transmitido (intervalo depende do sistema operacional)
Registrador do último parâmetro (flutuante)	Registro onde o último parâmetro flutuante das chamadas de funções C é transmitido (intervalo depende do sistema operacional)
Registrador para valores de retorno (flutuante)	Registro onde os parâmetros flutuantes das chamadas de funções C são retornados (intervalo depende do sistema operacional)
Ordem de byte INTEL	Modo de endereçamento de byte Intel é usado
Espaço máximo na pilha para argumentos (Byte)	Tamanho da pilha para argumentos (em bytes). Padrão: 40

Tabela 7-26. Diálogo Configurações do dispositivo 'PowerPC', Plataforma do Dispositivo

Sistema do dispositivo MIPS, Categoria Plataforma do Dispositivo

Diálogo Configurações do dispositivo 'MIPS', Plataforma do Dispositivo

Item do diálogo	Significado
Plataforma	Tipo do dispositivo
Registrador do primeiro parâmetro (inteiro)	Registro onde o primeiro parâmetro inteiro das chamadas de funções C é transmitido (intervalo depende do sistema operacional)
Registrador do último parâmetro (inteiro)	Registro onde o último parâmetro inteiro das chamadas de funções C é transmitido (intervalo depende do sistema operacional)
Registrador para valores de retorno (inteiro)	Registro onde o parâmetro inteiro das chamadas de funções C é retornado (intervalo depende do sistema operacional)
Espaço máximo na pilha para argumentos (Byte)	Depende do Sistema Operacional: tamanho máximo (número de bytes) dos argumentos, que podem ser entregues sobre a pilha

Tabela 7-27. Diálogo Configurações do dispositivo 'MIPS', Plataforma do Dispositivo

Sistema do dispositivo 'Hitachi SH', Categoria Plataforma do Dispositivo

Diálogo Configurações do dispositivo 'Hitachi SH', Plataforma do Dispositivo

Item do diálogo	Significado
Plataforma	Tipo do dispositivo
Registrador do primeiro parâmetro (inteiro)	Registro onde o primeiro parâmetro inteiro das chamadas de funções C é transmitido (intervalo depende do sistema operacional)
Registrador do último parâmetro (inteiro)	Registro onde o último parâmetro inteiro das chamadas de funções C é transmitido (intervalo depende do sistema operacional)
Registrador para valores de retorno (inteiro)	Registro onde o parâmetro inteiro das chamadas de funções C é retornado (intervalo depende do sistema operacional)
Espaço máximo na pilha para argumentos (Byte)	Depende do Sistema Operacional: tamanho máximo (número de bytes) dos argumentos, que podem ser entregues sobre a pilha
Suportar processador ponto flutuante	Comandos FPU são gerados para operações de ponto flutuante
Registrador do primeiro parâmetro (flutuante)	Registrador onde o primeiro parâmetro flutuante das chamadas de funções C é transmitido (intervalo depende do sistema operacional)
Registrador do último parâmetro (flutuante)	Registrador onde o último parâmetro flutuante das chamadas de funções C é transmitido (intervalo depende do sistema operacional)
Registrador para valores de retorno (flutuante)	Registro onde os parâmetros flutuantes das chamadas de funções C são retornados (intervalo depende do sistema operacional)

Ordem de byte INTEL	Modo de endereçamento de byte Intel é usado

Tabela 7-28. Diálogo Configurações do dispositivo 'Hitachi SH', Plataforma do Dispositivo

Sistema do dispositivo '8051 compatível, Categoria Plataforma do Dispositivo

Diálogo Configurações do dispositivo '8051 compatível', Plataforma do Dispositivo

Item do diálogo	Significado
Plataforma	Tipo do dispositivo
Tamanho máximo da pilha (em Bytes)	Tamanho máximo da pilha (número de Bytes)

Tabela 7-29. Diálogo Configurações do dispositivo '8051 compatível', Plataforma do Dispositivo

Tricore

As configurações do dispositivo para TriCore são codificadas por hardware e não são modificáveis via arquivo do dispositivo. Se, entretanto, modificações forem necessárias, por favor, contate a ALTUS.

Configurações do dispositivo na categoria Layout de memória

Os itens descritos para esta guia podem estar disponíveis para cada dispositivo padrão.

Diálogo Configurações do dispositivo..., Layout de Memória

Item do diálogo	Significado
Base (Código)	Automático ativado: segmento de código é alocado automaticamente Automático desativado: segmento de código se encontra no endereço absoluto fornecido
Base (Global)	Automático ativado: o segmento de dados (dados globais) é alocado automaticamente na área em questão Automático desativado: segmento de dados (dados globais) se encontra no endereço absoluto fornecido
Base (Memória)	Automático ativado: memórias são alocadas automaticamente para a área em questão Automático desativado: segmento de memórias se encontra no endereço absoluto fornecido
Base (Entrada)	Automático ativado: imagem do processo de entrada automaticamente alocada para a área em questão Automático desativado: imagem do processo de entrada se encontra no endereço absoluto fornecido
Base (Saída)	Automático ativado: imagem do processo de saída automaticamente alocada para a área em questão Automático desativado: imagem do processo de saída se encontra no endereço absoluto fornecido
Base (Retentivo)	Automático ativado: dados retentivos são alocados automaticamente para a área em questão Automático desativado: dados retentivos localizados no endereço absoluto fornecido
Área (Código)	Número do segmento do segmento de dados (código)
Área (Global)	Número do segmento dos segmentos de dados (dados globais)
Área (Memória)	Número do segmento do segmento de memória
Área (Entrada)	Número do segmento de imagem do processo de entrada
Área (Saída)	Número do segmento de imagem do processo de saída
Área (Retentivo)	Número do segmento dos dados retentivos
Tamanho (Código)	Tamanho do segmento de código
Tamanho pro Segmento(Global)	Tamanho do segmento de dados
Tamanho (Memória)	Tamanho do segmento de memória
Tamanho (Entrada)	Tamanho da imagem de processo de entrada
Tamanho (Saída)	Tamanho da imagem do processo de saída
Tamanho (Retentivo)	Tamanho do segmento de dados retentivos
Tamanho total da	Tamanho total da memória de dados

memória de dados	
Segmento retentivo próprio	Se ativado: dados retentivos são alocados em um segmento separado
Tamanho total da memória de dados	Tamanho total da memória de dados
Número máximo de segmentos de dados globais	Número máximo de segmentos de dados globais
Número máximo de POUs	Número máximo de POUs permitidas em um projeto

Tabela 7-30. Diálogo Configurações do dispositivo..., Layout de Memória

Configurações do dispositivo na categoria Geral

Os itens descritos para esta guia podem estar disponíveis para cada dispositivo padrão.

Diálogo Configurações do dispositivo, Geral

Item do diálogo	Significado
Configurável	Se ativado: suporta as configurações de E/S configuráveis e carrega a descrição de configuração para o controlador
Suporta configuração CANopen	Se ativado: suporta a configuração CANopen e carrega a descrição para o controlador
Suporta configuração Profibus	Se ativado: suporta a configuração Profibus e carrega a descrição para o controlador
Suporta multitarefa preemptiva	Se ativado: suporta a configuração de tarefas e carrega a descrição das tarefas para o controlador
Download como arquivo	Se ativado: a descrição de E/S é baixada no formato de arquivo
Sem verificação de endereço	Se ativado: na compilação os endereços IEC não são verificados
Alteração online	Se ativado: funcionalidade de alteração online
Atualizar E/S não usadas	Se ativado: o MasterTool IEC cria uma tarefa para atualizar as entradas e saídas que atualmente não são usadas. Assim, as E/S serão monitoradas na configuração do CP.
Tarefa única em multitarefa	Não foi implementado
Modo de endereçamento de byte	Se ativado: modo de endereçamento de bytes (ex.: var1 AT %QD4 é escrito no endereço %QB4)
Inicializar zero	Se ativado: inicialização geral com zero
Download de arquivo de símbolos	Se ativado: se um arquivo de símbolo tiver sido criado, ele será submetido a download
Configurar símbolos a partir de arquivo INI	Se ativado: os parâmetros para a configuração dos símbolos não são lidos a partir da caixa de diálogo de opções de projeto, mas do arquivo MasterTool IEC.ini, ou de outro arquivo referenciado no MasterTool IEC.ini
Navegador CP	Se ativado: funcionalidade do navegador CP ativada
Rastreamento	se ativado: funcionalidade Rastreamento ativada
VAR_IN_OUT como referência	Se ativado: em uma chamada de função, as variáveis VAR_IN_OUT são chamadas como referência (ponteiro); portanto, constantes não podem ser atribuídas e acesso de leitura/escrita não é possível fora do bloco funcional.
Inicializar entradas	Se não ativado: por motivos de otimização, nenhum código de inicialização será gerado para as entradas declaradas com "AT %IX" (-> valores indefinidos até o 1. ciclo de barramento!)
Carregar projeto de inicialização automaticamente	Se ativado: um projeto de inicialização é criado automaticamente após o download de um novo programa e enviado para o CP.
Softmotion	Se ativado: a funcionalidade SoftMotion é ativada, isto é, disponível na guia Recursos (lista de programa CNC, CAMs)
Forçamento retentivo	Se ativado: a Lista de forçamento será mantida no sistema em tempo de execução, mesmo em um logout. Nesse caso, o usuário receberá uma caixa de

	diálogo onde ele pode decidir se o forçamento realmente deve ser mantido (atualmente suportado em controladores MasterTool IEC SP 32 bits completo, V2.4, patch 4 e MasterTool IEC SP 32 bits).
Salvar	Se ativado: o controlador mantém o forçamento mesmo em uma reinicialização. Esta opção só estará disponível se permitido pelo dispositivo e se a opção 'Forçamento retentivo' (veja acima) está ativada.
Forçamento independente do ciclo	Se ativado: o forçamento não será feito apenas no início e no final de um ciclo, mas todos os acessos de escrita durante o processamento do programa serão desativados.

Tabela 7-31. Diálogo Configurações do dispositivo, Geral

Configurações do dispositivo na categoria Funcionalidade de rede

Os itens descritos para esta guia podem estar disponíveis para cada dispositivo padrão.

Diálogo Configurações do dispositivo - Funcionalidade de Rede

Item do diálogo	Significado
Suportar gerenciador de parâmetros	Se ativado: a entrada 'Gerenciador de parâmetros' aparece na Guia Recursos. Use-a para criar um dicionário de objetos para variáveis e parâmetros, que permitam ao dispositivo a troca de dados ativa com outros controladores
Suportar variáveis de rede	Se essa opção é selecionada, variáveis de rede podem ser usadas, o que permite a troca de dados automática na rede
Nomes de interfaces de rede suportadas	Lista de sistemas de rede suportados, por exemplo: CAN; UDP; DP
Intervalos de índice para parâmetros	Intervalo de índice para listas de parâmetros do tipo 'Parâmetros' (consultar Recursos, 'Gerenciador de parâmetros')
Intervalos de índice para variáveis	Intervalo de índice para listas de parâmetros do tipo 'Variáveis' (consultar Recursos, 'Gerenciador de parâmetros')
Intervalos de índice para mapeamentos	Intervalo de índice para listas de parâmetros do tipo 'Mapeamentos' (consultar Recursos, 'Gerenciador de parâmetros') Atenção: se esse intervalo é definido aqui, o CanDevice apenas considerará esse intervalo para o mapeamento; isso significa que se um intervalo de índice para Parâmetros adicional estiver definido (consultar acima) , aquele não será considerado!
Intervalo de sub-índice	Intervalo de sub-índice dentro dos intervalos de índice para dicionários de objeto de parâmetro e variável acima mencionados (consultar Recursos, ' Gerenciador de Parâmetros')

Tabela 7-32. Diálogo Configurações do dispositivo - Funcionalidade de Rede

Configurações do dispositivo na categoria Visualização

Os itens descritos para esta guia podem estar disponíveis para cada dispositivo padrão.

Diálogo Configurações do dispositivo, Visualização

Item do diálogo	Significado
Largura do display em pixel **Altura do display em pixel**	Uma área de determinada largura e altura será exibida na janela do editor ao editar uma visualização. Assim, por exemplo, o tamanho da tela em que a visualização do dispositivo será executada posteriormente, pode ser considerado ao posicionar os elementos de visualização.
Usar formato de arquivo 8.3	Os nomes dos arquivos de bitmaps e linguagem que são usados na visualização do MasterTool IEC serão reduzidos para o formato de notação 8.3 e carregados para o CP nesse formato.
Manipulação de alarmes no CP	A tarefa ALARM_TASK será inserida automaticamente na configuração da tarefa. Ela processará um código ST implicitamente criado avaliando o status dos alarmes específicos e, se aplicável, executando as ações associadas. O código ST precisa das funções auxiliares da biblioteca SysLibAlarmTrend.lib. Esta biblioteca será carregada automaticamente (adicionalmente, as implicitamente necessárias bibliotecas SysLibFile.lib SysLibSockets.lib, SysLibMem.lib, SysLibTime.lib são carregadas). Essas bibliotecas devem ser suportadas pelo Sistema do dispositivo! Se a opção é desativada e a visualização do dispositivo estiver ativada,

	um erro de compilação será gerado. Dica: o 'Manipulação de alarmes no CP' pode ser usado mesmo se nenhuma Visualização do dispositivo ou Web tiver sido ativada. Mesmo assim, o código ST necessário será gerado.
Armazenar dados de tendência no CP	A manipulação de tendência no CP será ativada. A tarefa TREND_TASK será inserida automaticamente na configuração da tarefa e processará um código ST implicitamente criado para gravação de dados de tendência em um buffer e - se a opção Histórico está ativada no elemento de tendências - para armazenar os valores em um sistema de arquivos. O código ST precisa das funções auxiliares da biblioteca SysLibAlarmTrend.lib. Esta biblioteca será carregada automaticamente (adicionalmente, as implicitamente necessárias bibliotecas SysLibFile.lib SysLibSockets.lib, SysLibMem.lib, SysLibTime.lib são carregadas). Essas bibliotecas devem ser suportadas pelo Sistema do dispositivo! Se a opção é desativada e a visualização do dispositivo estiver ativada, um erro de compilação será geado. Dica: 'Armazenar dados de tendência... ' pode ser usado mesmo se nenhuma Visualização do dispositivo ou Web tiver sido ativada. Mesmo assim, o código ST necessário será gerado.
Ativar variável de sistema 'CurrentVisu'	A variável do sistema CurrentVisu pode ser usada para alternar entre as visualizações
Fontes suportadas no dispositivo	Lista de fontes suportadas pelo sistema do dispositivo.
Manipulação de entradas simplificadas	Se ativado: no modo online a manipulação de entrada nas visualizações é simplificada: as teclas <Tab> e <espaço> não são necessárias para ir de um campo de entrada para outro. A seleção vai automaticamente para o próximo campo após ter encerrado a entrada com a tecla <Retornar>. Um campo de entrada também pode ser alcançado por meio das teclas de seta ou <Tab> e, então, imediatamente uma entrada pode ser inserida. Se não ativado: as teclas <Tab> e <espaço> devem ser usadas para ir para o próximo campo de entrada e para selecionar este campo para tornar possível uma entrada.
Visualização Web	Se ativado: todos os objetos de visualização do projeto são compilados para o uso como objetos de visualização Web.
Compressão	Se ativado: os seguintes arquivos para visualização Web, que devem ser transferido do MasterTool IEC para o Webserver/PLC, serão transferidos em um formato compactado (Formato- ZIP); caso contrário no formato original: Arquivos de visualização XML Arquivos de imagem (somente *.bmp, pois com outros não ocorre compactação) Arquivos de idioma (*.xml para textos dinâmicos, *.tlt, * .vis) Os arquivos além do nome do arquivo existente têm a extensão „.zip“. O ponto no nome existente será substituído por um sublinhado (exemplo: „ PLC_VISU.xml “ será renomeado para „PLC_VISU_xml.zip “) Nenhuma compactação é feita para a arquivos Java (minml.jar, webvisu.jar) e para a página principal webvisu.htm.
Prevenir download de arquivos de visualização	Se ativado: quando ocorre download do projeto, todos os arquivos que são usados na visualização atual não serão baixados para o sistema do dispositivo. Arquivos de visualização somente são baixados para visualização do dispositivo ou web e podem ser bitmaps, arquivos de idioma e para visualização Web, bem como arquivos de descrição XML.
Funções do teclado para tabelas	Se esta opção estiver ativada no modo online, as funções do teclado de tabelas na visualização (MasterTool IEC HMI, visualização Web, visualização do dispositivo) são possíveis. Desativando essa opção fará com que nenhum código seja gerado para as funções principais, o que pode ser razoável por motivos de desempenho ao usar a visualização do dispositivo.
Visualização do dispositivo	Se ativado: todos os objetos de visualização do projeto são compilados para o uso como objetos de visualização do dispositivo.
Usar VISU_INPUT_TASK	(podem apenas ser ativados caso a visualização do dispositivo esteja ativada, veja acima) Se ativado e 'Desativar a geração de tarefas' (veja abaixo) está desativada, então automaticamente duas tarefas serão criadas para controlar a Visualização do dispositivo: VISU_INPUT_TASK controla o processamento das entradas do usuário por meio da POU implicitamente disponível. MAINTARGETVISU_INPUT_CODE VISU_TASK controla o redesenho dos elementos de visualização por meio

	do POU implicitamente disponível. MAINTARGETVISU_PAINT_CODE. Se a opção estiver desativada, somente VISU_TASK será criado e apenas a POU MAINTARGETVISU_PAINT_CODE POU é usada, que nesse caso incluirá também a funcionalidade de MAINTARGETVISU_INPUT_CODE.
Desativar geração de tarefas	(podem apenas ser ativados caso a visualização do dispositivo esteja ativada, veja acima) Se a opção estiver ativada, as tarefas VISU_INPUT_TASK e VISU_TASK (consultar acima 'Uso do VISU_INPUT_TASK') não serão geradas automaticamente. Assim, das POUs mencionadas acima e, se a opção 'Usar VISU_INPUT_TASK' não for ativada, somente MAINTARGETVISU_PAINT_CODE POU pode ser chamada no programa de aplicativo e, se desejar, pode ser controlada por qualquer tarefa, conforme definido pelo programador. Sobre isto consultar a descrição na visualização do dispositivo.

Tabela 7-33. Diálogo Configurações do dispositivo, Visualização

Apêndice H: funções do teclado

Funções do teclado

Para executar o Master Tool IEC usando apenas o teclado, poderá ser necessário usar alguns comandos que não são encontrados no menu.

A tecla de função <F6> permite alternar e desfazer dentro da POU aberta entre as partes de declaração e as de instrução.

<Alt>+<F6> permite mover de um objeto aberto para o Organizador de objetos e daí para a janela de mensagens se ela estiver aberta. Se uma caixa de Pesquisa está aberta,<Alt>+<F6> permite que se alterne do Organizador de objetos para a caixa Pesquisar e, de lá, para o objeto.

Pressione <Ctrl>+<F6> para mover para a próxima janela do editor aberta, pressione <Ctrl>+<Shift>+<F6> para obter o anterior.

Pressione <Tab> para mover através dos campos de entrada e botões nas caixas de diálogo.

As teclas de seta permitem-lhe percorrer os cartões de registro e os objetos no âmbito do Organizador de Objetos e o Gerenciador de Bibliotecas.

Todas as outras ações podem ser executadas usando os comandos do menu ou com os atalhos listados após os comandos do menu.<Shift>+<F10> abre o menu de contexto que contém os comandos mais freqüentemente utilizados para o objeto selecionado ou para o editor ativo.

Combinações de teclas

Esta é uma visão geral de todas as combinações de teclas e teclas de função:

Funções gerais	
Mover entre a parte de declaração e a parte de instrução de uma POU	<F6>
Menu de contexto	<Shift>+<F10>
Modo de atalho para as declarações	<Ctrl>+<Enter>
Mover de uma mensagem na janela de mensagens para a posição inicial no editor	<Enter>
Mover para a próxima janela aberta do editor	<Ctrl>+<F6>
Mover para a janela do editor aberta anteriormente	<Ctrl>+<Shift>+<F6>
Abrir e fechar variáveis multi-camadas	<Enter>
Abrir e fechar pastas	<Enter>
Alternar cartões de registro no Organizador de objetos e no Gerenciador de Bibliotecas	<Setas>
Mover para o próximo campo em uma caixa de diálogo	<Tab>
Ajuda sensível ao contexto	<F1>

Tabela 7-34. Combinações de teclas e teclas de função (parte 1)

Comandos Gerais	
'Arquivo' 'Salvar'...	<Ctrl>+<S>
'Arquivo' 'Imprimir'	<Ctrl>+<P>
'Arquivo' 'Sair'	<Alt>+<F4>
'Projeto' 'Compilar'	<F11>
'Projeto' 'Excluir Objeto'	<Del>
'Projeto' 'Acrescentar Objeto'	<Ins>
'Projeto' 'Renomear Objeto'	<Espaço>
'Projeto' 'Abrir Objeto'	<Enter>
'Editar' 'Desfazer'	<Ctrl>+<Z>
'Editar' 'Refazer'	<Ctrl>+<Y>

'Editar' 'Recortar'	<Ctrl>+<X> ou <Shift>+<Del>
'Editar' 'Copiar'	<Ctrl>+<C>
'Editar' 'Colar'	<Ctrl>+<V>
'Editar' 'Excluir'	<Del>
'Editar' 'Localizar próximo'	<F3>
'Editar' 'Assistente de entrada'	<F2>
'Editar' 'Auto declarar'	<Shift>+<F2>
'Editar' 'Próximo erro'	<F4>
'Editar' 'Erro anterior'	<Shift>+<F4>
'Comunicação' 'Login'	<Alt><F8>
'Comunicação' 'Logout'	<Ctrl>+<F8>
'Comunicação' 'Run'	<F5>
'Comunicação' 'Alternar breakpoint'	<F9>
'Comunicação' 'Sobrepasso'	<F10>
'Comunicação' 'Passo a passo'	<F8>
'Comunicação' 'Ciclado'	<Ctrl>+<F5>
'Comunicação' 'Escrever valor(es)'	<Ctrl>+<F7>
'Comunicação' 'Forçar valor(es)'	<F7>
'Comunicação' 'Liberar Forçado(s)'	<Shift>+<F7>
'Comunicação' 'Diálogo escrever/forçar'	<Ctrl><Shift>+<F7>
'Janela' 'Mensagens'	<Shift>+<Esc>

Tabela 7-35. Combinações de teclas e teclas de função (parte 2)

Comandos do Editor FBD	
'Inserir' 'Rede (depois)'	<Ctrl>+<T>
'Inserir' 'Atribuição'	<Ctrl>+<A>
'Inserir' 'Salto'	<Ctrl>+<L>
'Inserir' 'Retorno'	<Ctrl>+<R>
'Inserir' 'Bloco funcional'	<Ctrl>+<B>
'Inserir' 'Entrada'	<Ctrl>+<U>
'Extras' 'Negar'	<Ctrl>+<N>
'Extras' 'Zoom'	<Alt>+<Enter>

Tabela 7-36. Combinações de teclas e teclas de função (parte 3)

Comandos do Editor CFC	
'Inserir' 'POU'	<Ctrl>+<B>
'Inserir' 'Entrada'	<Ctrl>+<E>
'Inserir' 'Saída'	<Ctrl>+<A>
'Inserir' 'Salto'	<Ctrl>+<J>
'Inserir' 'Rótulo'	<Ctrl>+<L>
'Inserir' 'Retorno'	<Ctrl>+<R>
'Inserir' 'Comentário'	<Ctrl>+<K>
'Inserir' 'Entrada de POU'	<Ctrl>+<U>
'Extras' 'Negar'	<Ctrl>+<N>
'Extras' 'Set/Reset'	<Ctrl>+<T>
'Extras' 'Conexão'	<Ctrl>+<M>
'Extras' 'EN/ENO'	<Ctrl>+<I>
'Extras' 'Zoom'	<Alt>+<Enter>

Tabela 7-37. Combinações de teclas e teclas de função (parte 4)

Comandos do Editor LD	
'Inserir' 'Rede (depois)'	<Ctrl>+<T>
'Inserir' 'Contato'	<Ctrl>+<K>
'Inserir' 'Contato (negado)'	<Ctrl>+<G>

'Inserir' 'Contato Paralelo'	<Ctrl>+<R>
'Inserir' 'Contato Paralelo (negado)'	<Ctrl>+<D>
'Inserir' 'Bloco funcional'	<Ctrl>+<B>
'Inserir' 'Bobina'	<Ctrl>+<L>
Inserir' 'Bobina 'SET''	<Ctrl>+<I>
'Inserir nos Blocos' 'Entrada'	<Ctrl>+<U>
'Inserir nos Blocos' 'Atribuição'	<Ctrl>+<A>
'Extras' 'Negar'	<Ctrl>+<N>
'Extras' 'Zoom'	<Alt>+<Enter>

Tabela 7-38. Combinações de teclas e teclas de função (parte 5)

Comandos do Editor SFC	
'Inserir' 'Transição de passo (antes)'	<Ctrl>+<T>
'Inserir' 'Transição de passo (depois)'	<Ctrl>+<E>
'Inserir' 'Ramificação alternativa (direita)'	<Ctrl>+<A>
'Inserir' 'Ramificação paralela (direita)'	<Ctrl>+<L>
'Inserir' 'Salto'	<Ctrl>+<U>
'Extras' 'Zoom ação/transição'	<Alt>+<Enter>
Mover de volta para o editor a partir da Visão Geral do SFC	<Enter>

Tabela 7-39. Combinações de teclas e teclas de função (parte 6)

Trabalhando com o CP e Configuração de Tarefas	
Abrir e fechar elementos de organização	<Enter>
Colocar uma caixa de controle de edição em volta do nome	<Espaço>
'Extras' 'Editar Entrada'	<Enter>

Tabela 7-40. Combinações de teclas e teclas de função (parte 7)

Trabalhando no Editor do Gerenciador de Parâmetros	
Alternar entre janela de navegação e editor da lista	<F6>
Excluir uma linha no editor da lista	<Ctrl>+<Del> <Shift>+<Del>
Excluir um campo no editor da lista	<Del>

Tabela 7-41. Combinações de teclas e teclas de função (parte 8)

Apêndice I: recomendações na nomeação de identificadores

Denominação de identificadores

Identificadores são definidos na declaração de variáveis (Nomes de variáveis), em tipos de dados definidos pelo usuário e na criação de POUs (funções, blocos funcionais, programas), assim como nas visualizações. Pode-se seguir as seguintes recomendações relativas à denominação de identificadores para torná-los exclusivos.

Identificadores para variáveis (nomes de variáveis)

A nomeação de variáveis em aplicativos e bibliotecas, tanto quanto possível, deve seguir a notação húngara.

Para cada variável, uma descrição significativa e curta deve ser associada (nome de base). A primeira letra de cada palavra de um nome de base deve ser uma letra maiúscula, as outras devem ser minúsculas (exemplo: FileSize). Se necessário, um arquivo de tradução para outros idiomas pode ser criado.

Antes do nome de base, correspondendo ao tipo de dados da variável, prefixos são adicionados em letras minúsculas.

Tipo de dados	Limite inferior	Limite superior	Bits	Prefixo	Comentário
BOOL	FALSE	TRUE	1 Bit	x*	
				b	Reservado
BYTE			8 Bit	by	Bit string, não usado para operações aritméticas
WORD			16 Bit	w	Bit string, não usado para operações aritméticas
DWORD			32 Bit	dw	Bit string, não usado para operações aritméticas
LWORD			64 Bit	lw	Não usado para operações aritméticas
SINT	-128	127	8 Bit	si	
USINT	0	255	8 Bit	usi	
INT	-32.768	32.767	16 Bit	i	
UINT	0	65.535	16 Bit	ui	
DINT	-2.147.483.648	2.147.483.647	32 Bit	di	
UDINT	0	4.294.967.295	32 Bit	udi	
LINT	-263	263 - 1	64 Bit	li	
ULINT	0	264 - 1	64 Bit	uli	
REAL			32 Bit	r	
LREAL			64 Bit	lr	
STRING				s	
TIME				tim	
TIME_OF_DAY				tod	
DATETIME				dt	
DATE				date	
ENUM			16 Bit	e	

POINTER				p	
ARRAY				a	

Tabela 7-42. Identificadores para variáveis

* apontado para variáveis BOOLeanas, x é escolhido como prefixo para diferenciar de BYTE e também para acomodar a percepção de um programador IEC (consultar endereçamento % IX0.0).

Exemplos:

```
BYSUBINDEX: BYTE;
SFILENAME: STRING;
UDICOUNTER: UDINT;
```

Em declarações aninhadas os prefixos são anexados uns aos outros na ordem das declarações:

Exemplo:

```
PABYTELEGRAMDATA: POINTER TO ARRAY [0..7] OF BYTE;
```

Instâncias de bloco funcional e **variáveis** de tipos de dados definidos pelo usuário como um prefixo obtém um atalho para o FB e o nome do tipo de dados (Exemplo: sdo).

Exemplo:

```
CANSDORECEIVEDTELEGRAM: CAN_SDOTELEGRAM;

TYPE CAN_SDOTELEGRAM :     (* PREFIX: SDO *)
STRUCT
   WINDEX:WORD;
   BYSUBINDEX:BYTE;
   BYLEN:BYTE;
   ABY: ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE
```

Constantes locais (c) iniciam com prefixo c e um caractere sublinhado, seguidos pelo prefixo do tipo e o nome da variável.

Exemplo:

```
VAR CONSTANT
   C_UISYNCID: UINT := 16#80;
END_VAR
```

Para variáveis globais (g) e constantes globais (gc) um prefixo adicional + sublinhado é anexado ao prefixo de biblioteca.

Exemplos:

```
VAR_GLOBAL
   CAN_G_ITEST: INT;
END_VAR
```

```
VAR_GLOBAL CONSTANT
   CAN_GC_DWEXAMPLE: DWORD;
END_VAR
```

Identificadores para tipos de dados definidos pelo usuário (DUT)

O nome de cada tipo de dados de estrutura consiste em um prefixo de biblioteca (exemplo: CAN), um sublinhado e uma descrição preferencialmente expressiva e curta (exemplo: SDOTelegram) da estrutura. O prefixo associado para as variáveis usadas dessa estrutura deve vir diretamente após os dois-pontos.

Exemplo:

```
TYPE CAN_SDOTELEGRAM :     (* PREFIX: SDO *)
STRUCT
   WINDEX:WORD;
   BYSUBINDEX:BYTE;
   BYLEN:BYTE;
   ABYDATA: ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE
```

Enumerações iniciam com o prefixo da biblioteca (exemplo: CAL), seguido de um sublinhado e do identificador em letras maiúsculas.

Considere que nas versões anteriores do MasterTool IEC, valores ENUM > 16 # 7FFF causaram erros, porque eles não foram convertidos automaticamente em valores INT. Por esse motivo ENUMs sempre devem ser definidos com valores INT corretos.

Exemplo:

```
TYPE CAL_DAY :(
CAL_MONDAY,
CAL_TUESDAY,
CAL_WEDNESDAY,
CAL_THURSDAY,
CAL_FRIDAY,
CAL_SATURDAY,
CAL_SUNDAY);
DECLARAÇÃO
ETODAY: CAL_DAY;
```

Identificadores para Funções, Blocos funcionais, Programas (POU)

Os nomes de funções, blocos funcionais e programas consistem no prefixo de biblioteca (exemplo: CAN), um sublinhado e um nome curto expressivo do POU (exemplo: SendTelegram). Como com as variáveis, sempre a primeira letra de uma palavra do nome do POU deve ser maiúscula, as outras devem ser letras minúsculas. É recomendável um verbo e um substantivo para compor o nome do POU.

Exemplo:

```
FUNCTION_BLOCK CAN_SENDTELEGRAM (* PREFIX: CANST *)
```

Na parte de declarações uma descrição resumida do POU deve ser fornecida como um comentário. Mais adiante, todas as entradas e saídas devem ser fornecidas com comentários. No caso de blocos funcionais o prefixo associado para configurar instâncias deve vir diretamente após o nome.

Ações não têm nenhum prefixo; somente aquelas ações que devem ser chamadas apenas internamente, isto é, pelo POU em si mesmo, iniciam com prv_.

Cada função - por motivo de compatibilidade com versões anteriores do MasterTool IEC - devem ter pelo menos um parâmetro. Funções externas não devem usar estruturas como valores de retorno.

Identificadores para visualizações

Nota: deve-se evitar que uma visualização tenha o mesmo nome de outra POU no projeto. Isso poderia levar a problemas no caso de alterações entre as visualizações.

Apêndice K: erros de compilação e advertências

Se forem detectados erros durante a compilação do projeto, mensagens aparecerão na janela de mensagens. Advertências também podem ser exibidas ali. <F4> sempre permite pular para a próxima linha da mensagem nesta janela, por meio da qual a POU referida também será aberta. Os erros e mensagens de advertência são precedidos por números exclusivos. Se uma linha de mensagem estiver marcada na janela de mensagens, <F1> abrirá a janela correspondente de ajuda online.

Advertências

1100

"Função desconhecida na biblioteca '<nome>'"

Uma biblioteca externa é usada. Por favor, verifique se todas as funções que são definidas no arquivo .hex, também são definidas no arquivo .lib.

1101

" Símbolo não resolvido '<Símbolo>'"

O gerador de código espera uma POU com o nome <Símbolo>. Ele não está definido no projeto. Defina uma função/programa com este nome.

1102

"Interface inválida para o símbolo '<Símbolo>'"

O gerador de código espera uma função com o nome < Symbol > e exatamente uma entrada escalar ou um programa com o nome < Symbol > e, não, uma entrada ou saída.

1103

"A constante '<nome>' no endereço de código '<endereço>' sobrescreve um limite de página de 16 k!"

Uma constante string excede o limite de página (16 k). O sistema não consegue processar isto. Depende do controlador se o problema poderia ter sido evitado por uma entrada no arquivo do dispositivo. Contate o fabricante do CP.

1200

"Tarefa '<nome>', chamada de '<nome>' Variáveis de acesso na lista parâmetro não estão atualizadas"

Variáveis que são usadas apenas em uma chamada de bloco funcional na configuração da tarefa, não serão listadas na lista de referência cruzada.

1300

"Arquivo não encontrado '<nome>'"

O arquivo, ao qual o objeto da variável global está apontando, não existe. Por favor, verifique o caminho.

1301

“Analyze-Library não encontrada! Código de análise não será gerado."

A função de análise é utilizada, mas a biblioteca analyze.lib está ausente. Adicione a biblioteca no Gerenciador de Bibliotecas.

1302

"Novas funções externas referenciadas inseridas. Por conseguinte, alterações online não são mais possíveis!"

Desde o último download foi vinculada uma biblioteca que contém funções ainda não referenciadas no controlador. Por esse motivo, deve-se baixar o projeto completo.

1400

"Pragma desconhecido '<nome>' é ignorado"

Este pragma não é suportado pelo compilador.Veja a palavra-chave ‘pragma’ para diretivas suportadas.

1401

"A estrutura '<nome>' não contém nenhum elemento."

A estrutura não contém quaisquer elementos, mas as variáveis desse tipo alocam 1 byte de memória.

1410

"'RETAIN' e 'PERSISTENT' não tem qualquer efeito nas funções"

Variáveis remanentes que são definidas localmente em funções são tratadas como as variáveis locais normais.

1411

"Variável "<name>' na configuração de variável não é atualizada em qualquer tarefa"

A instância de nível superior da variável não é referida por uma chamada em qualquer tarefa. Assim, não será copiada da imagem de processo.

Exemplo:

Configuração de variável:

```
VAR_CONFIG
 PLC_PRG.APRG.AINST.IN AT %IB0 : INT;
END_VAR
PLC_PRG:
INDEX := INDEXOF(APRG); ÍNDICE: = INDEXOF(APRG);
```

O programa aprg é referenciado, mas não é chamado. Assim, plc_prg.aprg.ainst.in nunca obterá o valor real de %IB0.

1412

"Token inesperado <nome> em pragma {nome do pragma}"

Se está usando um pragma que não está escrito corretamente e que não pode ser usado neste local. Consultar a palavra-chave 'pragma' na ajuda online do MasterTool IEC ou Guia de Usuários para obter ajuda para uma correção.

1413

"'<Nome>' não é uma chave válida para a lista '<Nome>'. A chave será ignorada"

No pragma uma lista de parâmetros inexistentes é especificada. Verifique o nome da lista e veja no Gerenciador de parâmetros quais as listas de atualmente disponíveis.

1414

Muitas definições de componentes em pragma '<nome>'

O pragma contém mais definições (em parênteses) do que elementos na matriz correspondente, bloco funcional ou estrutura.

1415

’ <Nome> ’ (<Número>): o '<número>' literal é atribuído a mais de uma enumeração

Na declaração da enumeração <Name> o número de mesmo é atribuído a mais de componentes de enumeração (por exemplo, TYPE aenum (A:= 1, b:= 1); END_TYPE).

1500

"A expressão não contém atribuição. Nenhum código foi gerado."

O resultado desta expressão não é usado. Por este motivo não existe código gerado para a expressão inteira.

1501

" Constante STRING passada como 'VAR_IN_OUT': '<Nome>' não deve ser sobrescrito! "

A constante não pode ser gravada dentro da POU, porque não há verificação de tamanho possível.

1502

" Variável '<Nome>' tem o mesmo nome que uma POU. A POU não será chamada!"

É usada uma variável que tem o mesmo nome de uma POU.

Exemplo:

```
PROGRAM A
...
VAR_GLOBAL
A: INT;
END_VAR
...
 A; (* A POU A NÃO É CHAMADA, MAS A VARIÁVEL A É CARREGADA. *)
```

1503

" A POU ‘<nome>’ não tem saídas. Resultado da caixa é definido como ‘TRUE’"

O pino de saída de uma POU que não tem saídas, está ligado no FBD ou LD. A atribuição automaticamente obtém o valor TRUE.

1504

" ’ <nome> ’ (‘ <número> ’): declaração não pode ser executada devido à avaliação da expressão lógica "

Eventualmente nem todas as ramificações da expressão lógica serão executadas.

Exemplo:

IF a AND funct(TRUE) THEN

Se a é FALSE então funct não será chamada.

1505

"Efeito colateral em '<Nome>'! Ramificação provavelmente não foi executada!"

A primeira entrada da POU é FALSE, por esta razão a ramificação lateral, que pode vir à segunda entrada, não será executada.

1506

"Variável '<nome>' tem o mesmo nome que uma ação local. A ação não será chamada!"

Renomeie a variável ou a ação.

1507

"Instância '<nome>' tem o mesmo nome que uma função. A instância não será chamada"

Foi chamada uma instância ST que tem o mesmo nome que uma função. A função será chamada! Use nomes diferentes.

1509

"'<nome>' ('<número>'): funções a serem registradas como retornos de chamada devem iniciar com 'callback'"

Se, ao usar uma função de retorno de chamada, seu nome não inicia com "callback", isto pode causar efeitos inesperados sobre os controladores RISC e Motorola 68K!

1550

"Várias chamadas da POU '<Nome>' em uma rede podem conduzir a efeitos colaterais indesejados"

Verifique se as chamadas múltiplas desta POU são realmente necessárias. Em caso de chamada múltipla valores indesejados podem ocorrer.

1600

"Abrir DB incorreto (código gerado pode estar incorreto). "

O programa original da Siemens não diz qual POU está aberta.

1700

"Entrada não conectada."

Uma caixa (entrada) que não tem nenhuma atribuição é usada no CFC. Por isso, nenhum código será gerado.

1750

" Passo '<Nome>': o tempo mínimo é maior que o tempo máximo! "

Abra a caixa de diálogo 'Atributos de passo' para este passo e corrija as definições de tempo.

1751

"Cuidado com o uso da variável '<nome>'. Esta variável é usada pelo código implícito e influencia o comportamento da seqüência de passo."

É recomendável renomear a variável, para que identificadores exclusivos sejam usados dentro do projeto e efeitos colaterais indesejados possam ser evitados.

1800

"<nome>(elemento #< número de elemento >): expressão de monitoração inválida '<name>' "

O elemento de visualização contém uma expressão que não pode ser controlada. Verifique o nome das variáveis e as substituições dos espaços reservados.

1801

"' <nome> (número): nenhuma entrada na expressão '<nome>' possível "

Na configuração da visualização do objeto em campo de entrada é usada uma expressão composta. Substitua-a por uma única variável.

1802

" <Objeto de visualização>(Número do elemento): bitmap '<nome>' não foi encontrado "

Certifique-se de que um arquivo de bitmap externo está disponível nesse caminho que é definido na caixa de diálogo de configuração da visualização .

1803

"'<nome>'('<número>'): "A ação de impressão não é suportada pelas visualizações Web e Do Dispositivo"

Uma ação de impressão é atribuída a um alarme configurado na visualização. Isso não será considerado na visualização Web ou do Dispositivo.

1804

"'<nome>'('<número>'): a fonte '<nome>' não é suportada pelo dispositivo. "

Na visualização está sendo usada uma fonte, que não é suportada pelo sistema do dispositivo. Consultar as configurações do dispositivo, na categoria 'visualização', as fontes suportadas.

1807

"<nome> (<número>): nenhuma janela de mensagens para alarmes na visualização do dispositivo"

Leve em conta que a ação "mensagem" não é suportada para a visualização do dispositivo!

1808

"<nome>'('<número>'): um polígono consiste de muitos pontos para visualização do dispositivo. Em caso de um elemento medidor, por favor, abra a configuração uma vez."

Por padrão, no máximo 512 pontos são permitidos, especificamente no dispositivo pode ser definido outro número máximo. Abrindo a caixa de diálogo de configuração, o elemento será otimizado ao número permitido de pontos.

1809

"'<nome>' ('<número>'): visualização inválida como destino de zoom: '<número>'"

Não é possível localizar esta visualização. Verifique se o nome de visualização não está correto ou o objeto de visualização não está disponível.

1850

"Variável de entrada em %IB <número> é usada na tarefa '<nome>' mas atualizada em outra tarefa"

Por favor, verifique quais tarefas estão usando esta variável e se a programação atual não está causando efeitos indesejáveis. A atualização do valor variável geralmente é feita na tarefa com a prioridade mais alta.

1851

"Variável de saída em %IQ<número> é usada na tarefa '<nome>' mas atualizada em outra tarefa"

Por favor, verifique quais tarefas estão usando esta variável e se a programação atual não está causando efeitos indesejáveis. A atualização do valor variável geralmente é feita na tarefa com a prioridade mais alta.

1852

"CanOpenMaster pode não ser chamado ciclicamente na tarefa de evento '<nome>'! Definir parâmetros do módulo UpdateTask!"

Atualmente, o CanOpen mestre é chamado pela tarefa de evento nomeada. Para chamá-lo ciclicamente, especifique uma tarefa apropriada, via parâmetros “Atualizar tarefas”, na configuração do CP, na caixa de diálogo 'Parâmetros do módulo'.

1853

" Um PDO (índice: '<número>') pode não ser atualizado ciclicamente na tarefa de evento '<nome>' "

Atualmente o PDO nomeado é controlado através do evento nomeado tarefa. Mas, para chamá-lo ciclicamente, deve-se atribuir uma tarefa apropriada ao PDO deslocando referências de E/S a esta tarefa.

1900

"POU "<nome>' (rotina principal) não está disponível na biblioteca"

A POU de início (PLC_PRG) não estará disponível, quando o projeto é usado como biblioteca.

1901

"Variáveis de acesso e configurações de variáveis não são salvas em uma biblioteca!"

Variáveis de acesso e a configuração de variável não são armazenadas na biblioteca.

1902

"’<Nome>’: não há nenhuma biblioteca para o tipo de máquina atual!"

O arquivo .obj da biblioteca foi gerado para outro dispositivo.

1903

"’<Nome>’: não é uma biblioteca válida"

O arquivo não tem formato solicitado para o dispositivo atual.

1904

"A constante '<Nome>' oculta uma constante do mesmo nome em uma biblioteca"

No projeto foi definida uma constante que tem o mesmo nome que o de outra, definida em uma biblioteca vinculada. A variável de biblioteca será sobrescrita!

1970

" Gerenciador de parâmetro: lista '<Nome>', coluna '<Nome>', valor '<Nome>' não pode ser importado!"

Verifique o arquivo de importação *.prm para entradas que não correspondem à configuração atual (valores padrão e descrição de arquivo XML) do Gerenciador de parâmetros.

1980

" Variáveis de rede global '<Nome>' '<Nome>: leitura e escrita simultânea pode resultar em perda de dados! "

Na configuração da lista de variáveis de rede (selecioe lista na guia Recursos e abra a caixa de diálogo 'Lista de variáveis globais', por meio do comando 'Propriedades' no menu de contexto),as opções 'leitura' e 'escrita' estão ativadas. Considere que isto pode resultar em perda de dados durante a comunicação.

1990

"Nenhuma 'VAR_CONFIG' para '<nome>' "

Para esta variável não existe nenhuma configuração de endereço disponível na Configuração de Variáveis (VAR_CONFIG). Abra a janela Configuração de Variáveis na guia Recursos e lá insira a configuração apropriada (comando 'Inserir' ’Todos caminhos de instância').

2500

"Tarefa '<nome da tarefa>': sem tempo de ciclo especificado para tarefa cíclica"

Na configuração da tarefa, uma tarefa cíclica foi criada, para a qual não foi definido um ciclo tempo. Insira um intervalo de tempo adequado na caixa de diálogo 'Atributos da tarefa' em "Intervalo".

Erros

3100

"Código grande demais. Tamanho máximo: '<número>' byte (<número> K) "

O tamanho de programa máximo foi excedido. Reduza o tamanho do projeto.

3101

"Total de dados muito grande. Tamanho máximo: '<número>' byte (<número> K)".

A memória foi excedida. Reduza o uso de dados do aplicativo.

3110

"Erro na biblioteca '<Name>'".

O arquivo. hex não é no formato hex da INTEL.

3112

"Instrução não realocável na biblioteca".

O arquivo .hex contém uma instrução que não é realocável. O código da biblioteca não pode ser vinculado.

3113

"O código da biblioteca sobrescreve as tabelas (função)".

Os intervalos para tabelas de código e função estão sobrepostas.

3114

"Biblioteca usa mais de um segmento".

As tabelas e o código no arquivo.hex usam mais de um segmento.

3115

" Não é possível atribuir constante a VAR_IN_OUT. Tipos de dados incompatíveis".

O formato de ponteiro interno para constantes string não pode ser convertido no formato de ponteiro interno de VAR_IN_OUT, porque os dados são definidos como "curtos" mas as constantes string são definidas como "longas". Se possível alterar essas configurações do dispositivo.

3116

Código 166x: a biblioteca externa não pode ser usada com as configurações atuais do dispositivo. Estes devem ser adaptados e a biblioteca deve ser reconstruída com configurações apropriadas.

3117

" <Nome> (<valor>): expressão muito complexa. Não existem mais registros disponíveis".

A expressão nomeada é muito complexa para ser tratada pelos registradores disponíveis. Por favor, tente reduzir a expressão usando variáveis intercalares.

3120

"Código do segmento atual excede 64 k".

O código gerado no momento é maior que 64 KB. Eventualmente, muito código de inicialização é criado.

3121

"POU grande demais".

Uma POU não pode exceder o tamanho de 64 k.

3122

"Inicialização grande demais. Tamanho máximo: 64 k".

O código de inicialização para uma função ou uma estrutura de POU não pode exceder 64 k.

3123

" Segmento de dados grande demais: segmento '<número> %s', tamanho <tamanho> bytes (máximo <número> bytes)".

Por favor, entre em contato com a ALTUS.

3124

"Constante string grande demais: '<número>' caracteres (máximo 253 caracteres)"

A constante fornecida deve ser reduzida no número de caracteres.

3130

“Pilha (usuário) pequena demais: '<número>' DWORD necessário, '<número>' DWORD disponível".

A profundidade de aninhamento das chamadas de POU é grande demais. Insira um tamanho de pilha maior nas configurações do dispositivo ou compile o projeto de compilação sem a opção ‚Depuração’ (pode ser definido no diálogo ‘ Projeto’ ‘Opções’ ‘Compilar’).

3131

"Pilha(usuária) pequena demais: '<número>' WORD necessário, '<número>' WORD disponível".

Por favor, contate a ALTUS.

3132

"Pilha (sistema) pequena demais : '<número>' WORD necessário, '<número>' WORD disponível".

Por favor, contate a ALTUS.

3150

" Parâmetro <número> da função '<nome>': não é possível passar o resultado de uma função IEC como parâmetro string para uma função C".

Use uma variável intermediária, ao qual o resultado da função IEC está atribuído.

3160

"Não é possível abrir o arquivo de biblioteca '<nome>' ".

Uma biblioteca <nome> está incluída no Gerenciador de Bibliotecas para este projeto, mas o arquivo da biblioteca não existe no caminho determinado.

3161

"A biblioteca '<nome>' não contém código de segmento".

Um arquivo.obj de uma biblioteca deve conter, no mínimo, uma função C. Insira uma função dummy no arquivo.obj, que não é definido no arquivo .lib.

3162

" Não pôde resolver a referência na biblioteca ' <nome> '(Símbolo '<nome>', Classe '<nome>', Tipo '<nome>')".

O arquivo.obj contém uma referência que não pode ser resolvida para outro símbolo. Por favor, verifique as configurações do compilador C.

3163

"Tipo de referência desconhecido na biblioteca '<nome>' (símbolo '<nome>', classe '<nome>' tipo '<nome>')"

O arquivo.obj contém um tipo de referência que não pode ser resolvido pelo gerador de código. Por favor, verifique as configurações do compilador C.

3200

" <nome>: expressão booleana muito complexa"

A memória temporária do sistema do dispositivo é insuficiente para o tamanho da expressão. Divida a expressão em várias expressões partes, usando atribuições para variáveis intermediarias.

3201

"<nome> (<rede>): uma rede não deve resultar em mais de 512 bytes de código"

Saltos internos não podem ser resolvidos. Ative a opção "Usar offsets de salto de16 bits" nas configurações do dispositivo 68 k.

3202

"Pilha saturada com chamadas de função de string/ matriz/estrutura aninhadas"

Uma chamada de função aninhada CONCAT(x, f(i)) é usada. Isso pode levar à perda de dados. Divida a chamada em duas expressões.

3203

"Expressão muito complexa (muitos endereços de registradores usados)"

Divida a atribuição em várias expressões.

3204

"Um salto excede 32 k bytes"

Distâncias de salto não podem ser maiores que 32767 bytes.

3205

" Erro interno: muitas constantes string"

Em um POU podem ser utilizadas no máximo 3.000 constantes string.

3206

"Dados do bloco funcional excedem tamanho máximo".

Um bloco funcional pode produzir um código de no máximo 32767 bytes.

3207

" Otimização de Matriz"

A otimização de acessos de matriz falhou porque, durante o cálculo do índice, uma função foi chamada.

3208

"Conversão ainda não implementada"

É usada uma função de conversão que não foi implementada para o gerador de código atual.

3209

"Operador não implementado"

É usado um operador, que não foi implementado para este tipo de dados e para o gerador de código atual. MIN(string1,string2).

3210

"Função '<Nome>' não encontrada"

É chamada uma função que não está disponível no projeto.

3211

"Uso máx da string excedido"

Uma variável do tipo string pode ser usada em uma expressão no máximo 10 vezes.

3212

"Ordem errada de biblioteca na POU < POU nome >"

A ordem das bibliotecas para esta POU não corresponde com a do arquivo cslib.hex. Corrija a ordem em conformidade (apenas para dispositivos 68K, se a opção de verificação estiver ativada no arquivo do dispositivo).

3250

"REAL não suportada para controlador de 8 bit "

O dispositivo atual não é suportado.

3251

" Tipos de hora do dia não são suportados para controlador de 8 bit "

O dispositivo atual não é suportado.

3252

"Tamanho da pilha excede <número> bytes"

O dispositivo atual não é suportado.

3253

" Não pôde localizar arquivo hex: '<Nome>' "

O dispositivo atual não é suportado.

3254

"Chamada de função de biblioteca externa não pôde ser resolvida."

O dispositivo atual não é suportado.

3255

"Ponteiros não são suportados para controladores de 8 bits."

Evite usar ponteiros no seu programa em sistema de 8 bits.

3260

"Função , name>’ tem muitos argumentos: aumentar o tamanho da pilha de argumentos nas configurações do dispositivo"

Se possível, modifique o tamanho da pilha no diálogo ‚Plataforma do Dispositivo’ nas configurações do dispositivo (padrão: 40). Se a configuração não for editável no MasterTool IEC, por favor, contate a ALTUS.

3400

"Ocorreu um erro durante a importação de variáveis de acesso"

O arquivo.exp contém uma seção incorreta de variáveis de acesso.

3401

"Ocorreu um erro durante a importação da configuração da variável "

O arquivo.exp contém uma seção incorreta de variáveis de configuração.

3402

"Ocorreu um erro durante a importação das variáveis globais"

O arquivo.exp contém uma seção incorreta de variáveis globais.

3403

"Não foi possível importar <nome>"

A seção para o objeto <nome> no arquivo .exp não está correta.

3404

"Ocorreu um erro durante a importação da configuração da tarefa"

A seção para configuração da tarefa no arquivo.exp não está correta.

3405

"Ocorreu um erro durante a importação de configuração do CP"

A seção para a configuração do CP no arquivo.exp não está correta.

3406

" Dois passos com o nome '<nome>'. Segundo passo não importado. "

A seção para a POU SFC no arquivo.exp contém dois passos com nomes iguais. Renomear um dos passos na exportação do arquivo.

3407

" Passo predecessor '<nome>' não encontrado"

O passo <nome> está ausente no arquivo.exp..

3408

" Passo sucessor '<nome>' não encontrado"

O passo <nome> está ausente no arquivo.exp..

3409

"Nenhuma transição sucede o passo '<nome>' "

No arquivo .exp uma transição está ausente, o que requer o passo <nome> como passo anterior.

3410

"Nenhum passo sucede a transição '<nome>' "

No arquivo.exp, um passo está ausente, o que requer que a transição <nome> o preceda.

3411

"Passo '<nome>' não acessível a partir do passo inicial"

No arquivo.exp a conexão entre passo <nome> e o passo inicial está ausente.

3412

"A macro '<nome>' não foi importada"

Verifique o arquivo de exportação.

3413

"Erro durante a importação dos CAMs."

Foi importado um arquivo de exportação (*.exp) que contém informações erradas sobre uma CAM. Verifique o arquivo de exportação.

3414

"Erro durante a importação da lista de programa CNC"

Foi importado um arquivo de exportação (*.exp) que contém informações erradas sobre um programa CNC. Verifique o arquivo de exportação.

3415

Erro durante a importação da configuração do alarme"

Foi importado um arquivo de exportação (*.exp) que contém informações erradas sobre configuração de alarme. Verifique o arquivo de exportação.

3450

" PDO '<PDO-nome>': ID do COB faltando! "

Clique no botão ‚Propriedades' no diálogo de configuração do CP para o módulo e digite uma ID para o PDO< Nome do PDO>.

3451

" Erro durante a carga: arquivo-EDS '<nome>' não pôde ser encontrado, mas é referenciado na configuração do hardware! "

Eventualmente, o arquivo do dispositivo necessário para a configuração de CAN não está no diretório correto. Verifique a configuração do diretório para arquivos de configuração em ‚Projeto' 'Opções' 'Diretórios'.

3452

"O módulo '<nome>' não pôde ser criado!"

O arquivo de dispositivo para o módulo <nome> não é adequado para a configuração atual. Eventualmente ele foi modificado desde que sua configuração foi definida no MasterTool IEC, ou ele está corrompido.

3453

"O canal '<nome>' não pôde ser criado!"

O arquivo de dispositivo para o canal <nome> não é adequado para a configuração atual. Eventualmente ele foi modificado desde que sua configuração foi definida no MasterTool IEC, ou ele está corrompido.

3454

"O endereço '<nome>' aponta para uma memória usada!"

A opção 'Verificar sobreposição de endereços' está ativada no diálogo ‚Configurações do CP' e uma sobreposição foi detectada. Considere que a área verificada é baseada no tamanho resultante dos tipos de dados dos módulos, e não no tamanho que é fornecido pela entrada ‚Tamanho' no arquivo de configuração.

3455

" Erro durante a carga: arquivo-GSD '<nome>' não pôde ser encontrado, mas é referenciado na configuração do hardware! "

Eventualmente, o arquivo de dispositivo solicitado para a configuração Profibus não está no diretório correto. . Verifique a configuração do diretório para arquivos de configuração em ‚Projeto' 'Opções' 'Diretórios'.

3456

"O dispositivo Profibus '<nome>' não pôde ser criado!"

O arquivo de dispositivo para o módulo <nome> não é adequado para a configuração atual. Eventualmente ele foi modificado desde que sua configuração foi definida no MasterTool IEC, ou ele está corrompido.

3457

"Erro na descrição do módulo!"

Por favor, verifique o arquivo do dispositivo deste módulo.

3458

"A configuração do CP não pôde ser criada! Verifique os arquivos de configuração".

Verifique se todas as configurações necessárias e arquivos de dispositivos estão disponíveis no caminho correto (ver diretório de compilação definido em 'Projeto' / 'Opções' ‘Diretórios’).

3459

"A velocidade de transmissão selecionada não é suportada."

Alterar a configuração no diálogo de Parâmetro CAN. Considere a especificação da taxa de baudrate fornecida pelo arquivo GSD.

3460

"3S_CanDrv.lib tem a versão errada"

Certifique-se de que o 3S_CanDrv.lib que está incluído no projeto, esteja atualizado.

3461

"3S_CanOpenMaster.lib tem a versão errada."

Certifique-se de que o 3S_CanOpenMaster.lib que está incluído no projeto, esteja atualizado.

3462

"3S_CanOpenDevice.lib tem a versão errada."

Certifique-se de que o 3S_CanOpenDevice.lib que está incluído no projeto esteja atualizado.

3463

"3S_CanOpenManager.lib tem a versão errada."

Certifique-se, de que o 3S_CanOpenManager.lib que está incluído no projeto esteja atualizado.

3464

"3S_CanNetVar.lib tem a versão errada."

Certifique-se de que o 3S_CanNetVar.lib que está incluído no projeto esteja atualizado.

3465

" CanDevice: sub índices devem ser numerados sequencialmente "

Nas listas de parâmetro utilizadas pela CanDevice, os índices devem ser numerados seqüencialmente e sem interrupção. Verifique a lista correspondente no Gerenciador de Parâmetros.

3466

" Variáveis de rede CAN: Controlador CAN não encontrado na configuração do CP"

Existem variáveis de rede configuradas para uma rede CAN (recursos, variáveis globais), mas na configuração do CP não há nenhum controlador CAN disponível.

3468

" CanDevice: tarefa de atualização não disponível na configuração da tarefa. "

A tarefa de atualização (usada para chamar o CANdevice), que está definida na caixa de diálogo de configuração do dispositivo CAN no CP, deve ser configurada na Configuração de Tarefas do projeto.

3469

"O CanOpenMaster não pode ser chamado. Por favor, atribua uma tarefa manualmente."

Atribua uma tarefa, a qual deve chamar o mestre, via parâmetro UpdateTask, na caixa de diálogo Parâmetros do módulo na Configuração do CP.

3470

"Nome inválido no parâmetro UpdateTask"

Abra a caixa de diálogo de parâmetro de módulo do CanMasters na configuração do CP. Verifique o parâmetro UpdateTask. A tarefa especificada deve estar disponível no projeto. Se não pode ser definida uma tarefa adequada aqui, o arquivo de dispositivo deve ser verificado para as definições de valor correspondentes para UpdateTask.

3500

"Sem 'VAR_CONFIG' para '<Nome>' "

Insira uma declaração para essa variável na lista de variáveis globais que contenha a 'Variable_Configuration'.

3501

"Sem endereço em 'VAR_CONFIG' para '<nome>'."

Atribuia um endereço a essa variável na lista de variáveis globais que contém a 'Variable_Configuration'.

3502

" Tipo de dados errados para ' <nome> em VAR_CONFIG "

Na lista de variáveis globais que contém a ‚Variable_Configuration ’ a variável é declarada com um tipo de dados diferente do que na POU.

3503

" Tipo de dados errados para '<nome>' em 'VAR_CONFIG' "

Na lista de variáveis globais que contém a ‚Variable_Configuration ’ a variável é declarada com um endereço diferente do que na POU.

3504

" Valores iniciais não são suportados para 'VAR_CONFIG’ "

Uma variável da ‚Variable_Configuration’ é declarada com endereço e valor inicial. Mas um valor inicial só pode ser definido para variáveis de entrada sem atribuição de endereço.

3505

" ’<nome>’ não é nenhum caminho de instância válida "

A Variable_Configuration contém uma variável inexistente.

3506

"Caminho de acesso esperado"

Na lista de variáveis globais para variáveis de acesso, o caminho de acesso para uma variável não está correto. Corrigir: <Identificador>:'< caminho de acesso >':<Tipo> <Modo de acesso>.

3507

" Sem especificação de endereço para variáveis 'VAR_ACCESS'"

A lista de variáveis globais para Variáveis de Acesso contém uma atribuição de endereço para uma variável. Isto não é permitido.

Definição de variável válida: <Identificador>:'< Caminho de acesso >':< Tipo > < Modo de Acesso >

3550

"Definição duplicada do identificador '<nome>'"

Existem duas tarefas que estão definidas com nomes idênticos. Renomeie uma delas.

3551

"A tarefa '<name>' deve conter pelo menos um chamada de programa"

Insira uma chamada de programa ou exclua a tarefa.

3552

"Variável de evento '<nome>' na tarefa '<nome>' não definida"

Existe uma variável de evento configurada no campo ‘Único’ da caixa de diálogo Propriedades da tarefa que não é declarada globalmente no projeto. Use outra variável, ou defina a variável globalmente.

3553

" Variável de Evento '<nome>' na tarefa '<nome>' deve ser do tipo 'BOOL' "

Use uma variável do tipo BOOL como variável de evento no campo ‘Único’ da caixa de diálogo propriedades de tarefa.

3554

"Entrada de tarefa '<nome>' deve ser um programa ou a instância global de Bloco funcional "

No campo ‚Chamada de programa’ é inserida uma função ou uma POU não definida. Digite um nome de programa válido.

3555

"A entrada de tarefa '<nome>' contém parâmetros inválidos"

No campo ‘Anexar chamada de programa’ existem parâmetros usados não conformes com a declaração do programa POU.

3556

"Tarefas não são suportadas pelo dispositivo atualmente selecionado"

A configuração da tarefa atualmente definida não pode ser usada para o sistema do dispositivo definido no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3557

"Número máximo de tarefas (<número>) excedido"

O número de tarefas atualmente definido excede o número máximo permitido para o sistema do dispositivo definido no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente. Atenção: não edite o arquivo de descrição XML da configuração de tarefa!

3558

" Prioridade da tarefa '<nome>' está fora do intervalo válido entre '< limite inferior >' e '< limite superior >" "

A prioridade atualmente definida para a tarefa não é válida para o sistema do dispositivo definido no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3559

" Tarefa '<nome>': tarefa de intervalo não é suportada pelo dispositivo atual "

A configuração da tarefa atual contém uma tarefa de intervalo. Isto não é permitido pelo sistema do dispositivo definido no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3560

" Tarefa '<nome>': tarefas de execução livre não são suportadas pelo dispositivo atual "

A configuração da tarefa atual contém uma tarefa de execução livre. Isto não é permitido pelo sistema do dispositivo definido no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3561

" Tarefa '<nome>': tarefas eventos não são suportadas pelo dispositivo atual "

A configuração atual da tarefa contém eventos que não são suportados pelo sistema do dispositivo definido no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3562

Tarefa '<nome>': tarefas de eventos externos não são suportadas pelo dispositivo atual "

A configuração atual da tarefa contém tarefas de evento externo que não são suportadas pelo sistema do dispositivo definido no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3563

" O intervalo da tarefa '<nome>' está fora do intervalo válido entre '< limite inferior >' e '< limite superior >' "

Altere o valor do intervalo na caixa de diálogo Configuração para a tarefa.

3564

"O evento externo '<nome>' da tarefa '<nome>' não é suportado pelo dispositivo atual"

O sistema do dispositivo definido no momento não suporta o evento externo que é definido na configuração da tarefa para esta tarefa. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3565

Número máximo de tarefas de eventos (<número>) excedido"

O sistema do dispositivo definido no momento não permite tantas tarefas de evento quantas definidas no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3566

"Número máximo de tarefas de intervalo (<número>) excedido"

O sistema do dispositivo definido no momento não permite tantas tarefas de intervalo quantas definido no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3567

"Número máximo de tarefas de execução livre (<número>) excedido"

O sistema do dispositivo definido no momento não permite, tantas tarefas de execução livre quantas definidas no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3568

"Número máximo de tarefas de intervalo externo (<número>) excedido"

O sistema do dispositivo definido no momento não permite tantos intervalos de tarefas externas quantos definidos no momento. Altere o dispositivo ou modifique a configuração de tarefa correspondente.

3569

"POU '<nome>' para evento de sistema '<nome>' não definida"

A POU que deve ser chamada pelo evento de sistema nomeado, tal como definido na configuração da tarefa, não está disponível no projeto. Modifique a configuração da tarefa correspondente ou verifique se a POU está disponível no projeto.

3570

"As tarefas '<nome>' e '<nome>' partilham a mesma prioridade"

Modifique a configuração de tarefa para que cada tarefa tenha uma prioridade diferente.

3571

"A biblioteca 'SysLibCallback' não está incluída no projeto! Eventos do sistema não podem ser gerados."

Para criar tarefas de eventos, é necessário o SysLibCallback.lib. Vincule esta biblioteca ao projeto no Gerenciador de Bibliotecas ou modifique a configuração de tarefa (atributos da tarefa) de forma que não haja nenhuma tarefa acionada por um evento.

3572

"Intervalo de cão de guarda de tarefa '<nome>' está fora do intervalo válido de '<número> µs' para '<número> µs' "

Na configuração da tarefa de diálogo 'Atributos de tarefa' existe um tempo de cão de guarda definido em microssegundos, que está fora do intervalo válido definido no arquivo de descrição XML.

3573

" Intervalo de cão de guarda de tarefa '<nome> ' está fora do intervalo válido de <valor> %' "

Na configuração da tarefa no diálogo 'Atributos de tarefa' existe um tempo de cão de guarda definido em percentual, que está fora do intervalo válido definido no arquivo de descrição XML.

3574

" A variável de evento '<nome>' e, respectivamente, seu endereço direto não devem ser usados várias vezes como um evento"

Um evento de Singleton é usado várias vezes na configuração da tarefa. Para obter informações sobre eventos de singleton consultar a descrição em 'Inserir' 'Inserir tarefa' ou 'Inserir' 'Anexar tarefas'.

3575

" Tarefa '<nome>': o tempo de ciclo deve ser um múltiplo de '<número>' µs "

Corrija o tempo de ciclo na caixa de diálogo Atributos de tarefa para esta tarefa. O sistema do dispositivo define um tempo de base e prescreve que o tempo de ciclo deve ser igual ou ser um múltiplo deste.

3600

"Variáveis implícitas não encontradas!"

Use o comando ,'Compilar tudo'. Se, no entanto, obtiver a mensagem de erro novamente, contate a ALTUS.

3601

" <nome> é um nome de variável reservada"

A variável fornecida é declarada no projeto, embora este seja reservado para o gerador de código. Renomeie a variável.

3610

“'<Nome>' não suportado"

O recurso fornecido não é suportado pela versão atual do sistema de programação.

3611

"O diretório de compilação fornecido '<nome>' é inválido"

Existe um diretório inválido fornecido em ‚Projeto’ ‚Opções’ ‚Diretórios’ para os arquivos de compilação.

3612

"Número máximo de POUs (<número>) excedido! Compilação foi abortada."

Muitos tipos de POUs e tipos de dados são usados no projeto. Modifique o número máximo de POUs nas configurações do dispositivo/layout de memória.

3613

“Compilação cancelada"

O processo de compilação foi cancelado pelo usuário.

3614

"Projeto deve conter uma POU chamada '<nome>' (rotina principal) ou uma configuração de tarefa"

Crie uma POU de inicialização do tipo de Programa (por exemplo, PLC_PRG) ou crie uma configuração de tarefa.

3615

"<Nome> (rotina principal) deve ser do tipo Programa"

É usada no projeto uma inicialização POU (por exemplo, PLC_PRG) que não é do tipo Programa.

3616

"Programas não devem ser implementados em bibliotecas externas"

O projeto que deve ser salvo como uma biblioteca externa contém um programa. Ele não estará disponível quando a biblioteca for utilizada.

3617

"Sem memória"

Aumente a capacidade de memória virtual do computador.

3618

"Acesso a bit não suportado no gerador de código atual!"

O gerador de código para o sistema do dispositivo definido no momento não suporta acesso a bit em variáveis.

3619

"Arquivo de objeto '<nome>' e biblioteca '<nome>' têm versões diferentes!"

Certifique-se de que estão disponíveis para a biblioteca versões correspondentes de arquivos *.lib, *.obj e *.hex . Esses arquivos devem ter o mesmo tempo de “stamp”.

3620

"A POU '<nome>' não deve estar presente no interior de uma biblioteca"

Deseja-se salvar o projeto como uma biblioteca na versão 2.1. Nesta versão uma biblioteca não pode conter um objeto PLC_PRG. Use um nome diferente de POU.

3621

"Não é possível gravar o arquivo de compilação '<nome>' "

Provavelmente, no caminho que é especificado para o arquivo de compilação já existe um arquivo de mesmo nome, o qual é de "somente leitura". Remova este arquivo e altere os direitos de acesso.

3622

"Não foi possível criar o arquivo de símbolos '<nome>' "

Provavelmente, no caminho que é especificado para o arquivo de símbolos (normalmente no diretório de projeto) já existe um arquivo de mesmo nome, o qual é de "somente leitura". Remova este arquivo e altere os direitos de acesso.

3623

" Não é possível gravar o arquivo de projeto de inicialização '<nome>' "

Provavelmente ,no caminho que é especificado para o arquivo (específico do dispositivo) já existe um arquivo de mesmo nome, o qual é de "somente leitura". Remova este arquivo e altere os direitos de acesso.

3700

" POU com nome ‘ <nome> 'já está na biblioteca' <nome> ' "

É usado no projeto um nome de POU que já foi usado para uma biblioteca POU. Renomeie a POU.

3701

"Nome usado na interface não é idêntico ao nome da POU"

Use o comando ‘Projeto’ ‘Renomear objeto’ para renomear a POU no Organizador de objetos, ou altere o nome da POU na janela declaração. Lá, o nome da POU deve ser colocado próximo a uma das palavras-chave PROGRAM, FUNCTION ou FUNCTIONBLOCK.

3702

"Estouro da lista do identificador"

Um máximo de 100 identificadores pode ser inserido em uma declaração de variável.

3703

"Definição do identificador '<Nome>' duplicada"

Providencie para que haja apenas um identificador com o nome fornecido na parte de declaração da POU.

3704

"Recursão de dados: <POU 0> -> <POU 1> -> .. -> <POU 0>"

É usada uma instância de um bloco funcional que chama a si mesmo.

3705

" <Nome>: VAR_IN_OUT em POU de nível superior não é permitido, se não houver nenhuma configuração de tarefa"

Crie uma configuração de tarefa ou certifique-se de que não há variáveis VAR_IN_OUT utilizadas em PLC_PRG.

3706

"Modificador 'CONSTANT' permitido somente para 'VAR', 'VAR_INPUT', 'VAR_EXTERNAL' e 'VAR_GLOBAL' "

Para essa categoria de variáveis, constantes não podem ser declaradas.

3720

"Endereço esperado após 'AT' "

Adicione um endereço válido após a palavra-chave AT, ou modifique a palavra-chave.

3721

"Apenas 'VAR' e 'VAR_GLOBAL' podem ser localizadas para endereços"

Ponha a declaração em uma área de declaração VAR ou VAR_GLOBAL.

3722

"Somente variáveis 'BOOL' são permitidas em endereços de bit"

Modifique o endereço ou modifique o tipo da variável à qual o endereço é atribuído.

3726

"Constantes não podem ser estabelecidas em endereços diretos"

Modifique a atribuição de endereço correspondente.

3727

"Não há declaração de matriz permitida neste endereço"

Modifique a atribuição de endereço correspondente.

3728

“Endereço inválido: '<endereço>' "

Este endereço não é suportado pela configuração do CP. Verifique a configuração do CP e modifique o endereço.

3729

" Tipo inválido '<nome>' no endereço: '<Nome>' "

O tipo desta variável não pode ser colocado no endereço fornecido. Exemplo: para um sistema do dispositivo trabalhando com ‘alinhamento 2’ a seguinte declaração não é válida: var1 AT % IB1: WORD;

Esta mensagem de erro também pode indicar que uma matriz é atribuída ao endereço de uma variável direta, o que não é permitido.

3740

" Tipo inválido: '<Nome>' "

Um tipo de dados inválido é usado em uma declaração de variável.

3741

"Esperando a especificação do tipo"

Uma palavra-chave ou um operador é usado em vez de um identificador de tipo válido.

3742

"Valor de enumeração esperado"

Na definição de tipo de enumeração um identificador está ausente após o parênteses de abertura ou após uma vírgula entre os parênteses.

3743

"Número inteiro esperado"

Enumerações só podem ser inicializadas com os números do tipo INT.

3744

"Enumeração (constante) '<nome>' já definida"

Verifique se foram seguidas as seguintes regras para a definição de valores de enumeração:

No âmbito de uma definição enum todos os valores devem ser exclusivos.

No âmbito de todas as definições globais de enum todos os valores devem ser exclusivos.

No âmbito todas as definições locais de enum todos os valores devem ser exclusivos.

3745

"Sub-intervalos só são permitidos em inteiros!"

Tipos de sub-intervalo só podem ser definidos sobre tipos de dados inteiro.

3746

"Sub-intervalo ‘<nome>' não é compatível com o tipo '<nome>' "

Um dos limites fixados para o intervalo do tipo sub-intervalo está fora do intervalo que é válido para o tipo de base.

3747

" Comprimento de string desconhecido: '<nome>' "

Não há uma constante válida, usada para a definição do comprimento da string.

3748

"Mais de três dimensões não são permitidas para matrizes"

Mais do que as três dimensões permitidas são fornecidas na definição de uma matriz. Se aplicável, use uma ARRAY OF ARRAY.

3749

"Limite inferior '<nome>' não definido"

Não há uma constante definida usada para definir o limite inferior para um tipo sub-intervalo ou matriz.

3750

"Limite superior '<nome>' não definido"

Não há uma constante definida usada para definir o limite superior para um tipo sub-intervalo ou matriz.

3751

"Comprimento da string inválido '<número de caracteres>' "

O comprimento da string aqui definida excede o valor máximo que é definido para o sistema do dispositivo definido no momento.

3752

"Mais de 9 de dimensões não são permitidas para matrizes aninhadas"

Uma matriz pode ser mono, bi ou tridimensional. O número de dimensões atingido pelo aninhamento de matrizes (por exemplo, "arr: ARRAY [0..2,0..2,0..2] OF ARRAY [0..2,0..2,0..2] OF ARRAY [0..2,0..2,0..2, 0..2] OF DINT" pode ser no máximo 9 e é excedido no caso do erro atual. Reduza adequadamente ao máximo de 9 dimensões.

3760

"Erro no valor inicial"

Use um valor inicial, que corresponde à definição de tipo. Para alterar a declaração que pode-se usar a caixa de diálogo de declaração para variáveis (Shift/F2 ou 'Editar' 'Auto-declarar').

3761

"Variáveis 'VAR_IN_OUT' não devem ter um valor inicial."

Remova a inicialização na declaração da variável VAR_IN_OUT.

3780

"'VAR', 'VAR_INPUT', 'VAR_OUTPUT' ou 'VAR_IN_OUT' esperadas"

A primeira linha após o nome de uma POU deve conter uma destas palavras-chave.

3781

"'END_VAR' ou identificador esperado"

Insira um identificador válido de um END_VAR no início da linha fornecida na janela de declaração.

3782

"Final inesperado"

No editor de declaração: adicione a palavra-chave END_VAR no final da parte da declaração.

No editor de texto da parte de programação: adicione uma instrução que termina a última seqüência de instruções (por exemplo, END_IF).

Esta mensagem de erro já pode ter sido criada juntamente com o erro 3703, se há duas declarações idênticas no final da parte da declaração.

3783

" END_STRUCT ' ou identificador esperado "

Assegure-se de que a declaração de tipo é encerrada corretamente.

3784

"O dispositivo atual não suporta o atributo < nome de atributo >"

O sistema do dispositivo não suporta este tipo de variáveis (por exemplo, RETENTIVAS, PERSISTENTES)

3800

"As variáveis globais precisam de muita memória. Aumente a memória disponível nas opções do projeto".

Aumentar o número de segmentos fornecidos nas configurações na caixa de diálogo ‚Projeto' ‚Opções' ‚Compilar '.

3801

"A variável '<Nome>' é muito grande (<tamanho> byte)"

A variável usa um tipo que é maior do que 1 segmento de dados. O tamanho do segmento é um parâmetro específico do dispositivo e pode ser modificado nas configurações do dispositivo/layout de memória. Se não encontrar isto nas configurações atuais do dispositivo, por favor, contate a ALTUS.

3802

"Sem memória retentiva. Variável '<nome>', '<número>' bytes"

O espaço de memória disponível para variáveis retentivas está esgotado. O tamanho da área de memória pode ser definido no dispositivo específico nas configurações do dispositivo/layout de memória. Se não encontrar o campo de configurações na caixa de diálogo, entre em contato com a ALTUS (considerar: se as variáveis retentivas são usadas em uma instância do Bloco funcional, a instância completa da POU será armazenada na área de memória retentiva!).

3803

"Sem memória de dados globais. Variável '<nome>', '<número>' bytes."

O espaço de memória disponível para variáveis globais está esgotado. O tamanho da área de memória pode ser definido no dispositivo específico nas configurações do dispositivo/layout de memória. Se não encontrar o campo de configurações na caixa de diálogo, entre em contato com a ALTUS.

3820

"'VAR_OUTPUT' e 'VAR_IN_OUT' não são permitidas em funções"

Em uma função, variáveis de saída ou entrada_saída não podem ser definidas.

3821

"Pelo menos uma entrada é necessária para funções"

Adicione, pelo menos, um parâmetro de entrada para a função.

3840

"Variável global desconhecida '<nome>'!"

Na POU, uma variável VAR_EXTERNAL é usada, para a qual nenhuma variável global foi declarada.

3841

"Declaração de '<nome>' não coincide com a declaração global!"

O tipo fornecido na declaração da variável VAR_EXTERNAL não é o mesmo que na declaração global.

3850

"Declaração de uma estrutura descompactada '<nome>' dentro de uma estrutura compactada '<nome>' não é permitida!"

Esta definição de estrutura conduz a um desalinhamento na memória. Deve-se alterar a definição de estrutura adequadamente.

3900

"Vários sublinhados no identificador"

Remova os vários sublinhados no nome do identificador.

3901

"No máximo, 4 campos numéricos são permitidos em endereços"

Há uma atribuição direta para um endereço que tem mais de quatro níveis (por exemplo, %QB0.1.1.0.1).

3902

"Palavras-chave devem ser maiúsculas"

Use maiúsculas para a palavra-chave ou ative a opção ‚Auto-formatar’ em ‚Projeto’ ‚Opções’.

3903

"Constante de tempo inválida"

A notação da constante não está em conformidade com o formato IEC61131-3.

3904

"Estouro na constante de tempo"

O valor utilizado para a constante de tempo não pode ser representado no formato interno. O valor máximo apresentável é t#49d17h2m47s295ms.

3905

"Constante de data inválida"

A notação da constante não está em conformidade com o formato IEC61131-3.

3906

" Constante de hora do dia inválida"

A notação da constante não está em conformidade com o formato IEC61131-3.

3907

" Constantes de data tempo e inválidas"

A notação da constante morre em não conformidade com o formato IEC61131-3.

3908

"Constante string inválida"

A constante string contém um caractere inválido.

4000

"Identificador esperado"

Insira um identificador válido nesta posição.

4001

"Variável '<Nome>' não declarada"

Declare a variável local ou global.

4010

"Incompatibilidade de tipo: não é possível converter '<Nome>' para '<Nome>'"

Verifique o tipo de dados que o operador espera (procure ajuda online para o nome do operador) e altere o tipo de variável que causou o erro ou selecione outra variável.

4011

" Incompatibilidade de tipo no parâmetro '<Nome>' de '<Nome>': não é possível converter '<Nome>' para '<Nome>'"

O tipo de dados no parâmetro atual não pode ser convertido automaticamente naquele parâmetro formal. Use uma conversão de tipo ou use outro tipo de variável.

4012

" Incompatibilidade de tipo no parâmetro '<Nome>' de '<Nome>': não é possível converter '<Nome>' para '<Nome>'"

Um valor com o tipo inválido <Typ2> é atribuído à variável de entrada '<Nome>'. Substitua a variável ou constante por uma do tipo <Typ1> ou use uma conversão de tipo, respectivamente, uma constante com prefixo de tipo.

4013

"Incompatibilidade de tipo na saída '<Nome>' de '<Nome>': não é possível converter '<Nome>' para '<Nome>'"

Um valor com o tipo inválido <Typ2> é atribuído à variável saída '<Nome>'. Substitua a variável ou constante por uma do tipo <Typ1> ou use uma conversão de tipo, respectivamente, uma constante com prefixo de tipo.

4014

" Literal digitado: não é possível converter '<nome>' em '<nome>' "

O tipo da constante não é compatível com o tipo do prefixo.

Exemplo: SINT#255

4015

" Tipo de dados ‘<nome>' ilegal para o acesso direto a bit "

Endereçamento de bit direto só é permitido para tipos de dados Inteiro e Cadeia de bits e de maneira nenhuma para variáveis diretas. Está sendo usada uma variável var1 do tipo REAL/LREAL ou uma constante em acesso de bit < var1 >.<bit>, ou tentando um acesso de bits em uma variável direta.

4016

"Índice de bits '<número>' fora do intervalo para variável do tipo '<nome>' "

Tentativa de acesso a um bit que não está definido para o tipo de dados da variável.

4017

"’MOD’ não está definido para 'REAL' "

O operador MOD só pode ser usado para tipos de dados Inteiro e Cadeia de bits.

4020

"Variável com acesso de escrita ou endereço direto requerido para , 'STN', 'S', 'R'"

Substitua o primeiro operando por uma variável com acesso de escrita.

4021

"Acesso de escrita à variável '<nome>' não permitido"

Substitua a variável por uma variável com acesso de escrita.

4022

“Operando esperado"

Adicione um operando atrás do comando.

4023

" Número esperado após '+' ou '-' "

Insira um dígito.

4024

"Esperando <Operador 0> ou <Operador 1> ou... antes de '<Nome>' "

Insira um operando válido na posição nomeada.

4025

" Esperando ':= ' ou ' => 'antes de'<Nome>' "

Introduza um dos dois operadores na posição nomeada.

4026

"'BITADR' espera um endereço de bit ou uma variável em um endereço de bit"

Usar um endereço de bit válido (por exemplo, %IX0.1).

4027

"Número inteiro ou constante simbólica esperada"

Insira um número inteiro ou o identificador de uma constante válida.

4028

"Operador 'INI' precisa de instância do bloco funcional ou instância de tipo de unidade de dados"

Verifique o tipo de dados da variável, para o qual o operador INI é usado.

4029

"Chamadas aninhadas da mesma função não são possíveis."

Uma chamada de função não pode conter uma chamada a si mesma como um parâmetro no modo simulação e em sistemas do dispositivo não reentrantes.

Exemplo: fun1(a,fun1(b,c,d),e);

Use uma tabela intermediária.

4030

"Expressões e constantes não são permitidos como operandos de "ADR"

Substitua a constante ou a expressão por uma variável ou um endereço direto.

4031

"’ADR' não é permitido em bits! Use 'BITADR’"

Use BITADR. Por favor, note: a função BITADR não retornará um endereço de memória física.

4032

"’<número>’ operandos são poucos para '<nome>'. Pelo menos ‘<número>’ são necessários"

" Verifique quantos operandos o operador nomeado requer e adicione os operandos faltantes.

4033

"'<número>' operandos são muitos para '<nome>'. Pelo menos '<número>' são necessários "

Verifique quantos operandos o operador nomeado requer e remova os operandos excedentes.

4034

"Divisão por 0"

Divisão por 0 em uma expressão constante. Para provocar um erro de tempo de execução, use, se aplicável, uma variável com o valor 0.

4035

"ADR não deve ser aplicado na 'VAR CONSTANT' se 'Substituír constantes' estiver ativada"

Não é possível um acesso de endereço nas constantes para as quais os valores diretos são usados. Se este for o caso, desative ‚Substituir constantes' em ‚Projeto','Opções','Compilar'.

4040

"Rótulo '<nome>' não está definido"

Defina um rótulo com o nome <NomeRótulo> ou altere o nome <NomeRótulo> para aquele definido no rótulo.

4041

"Definição duplicada do rótulo '<nome>' "

O rótulo '<Nome>' tem múltiplas definições na POU. Renomeie o rótulo ou remova uma das definições.

4042

"Não é permitido mais do que <número> rótulos na seqüência"

O número de rótulos de salto é limitado a '<valor>'. Inserir uma instrução dummy.

4043

"Formato do rótulo inválido. Um rótulo deve ser um nome opcionalmente seguido por dois pontos."

O nome do rótulo não é válido ou estão ausentes os dois pontos na definição.

4050

"POU '%s' não está definida"

Defina uma POU com o nome '<Nome>', usando o comando 'Projeto''Adicionar objeto' ou altere o '<Nome>' para o nome de uma POU definida.

4051

"'%s' não é nenhuma função"

Use, em vez de <Nome> , um nome de função que esteja definido do projeto ou nas bibliotecas.

4052

"'<nome>' deve ser uma instância declarada de FB '<name>'"

Use uma instância do tipo de dados '<Nome>' que é definida no projeto ou altere o tipo de <NomeInstância> para '<Nome>'.

4053

"'<nome>' não é uma caixa válida ou operador"

Substitua '<Nome>' pelo nome de uma POU ou um operador de definido no projeto.

4054

" Nome da POU esperada como parâmetro de 'INDEXOF'"

O parâmetro fornecido não é compatível com um nome POU válido.

4060

"Parâmetro 'VAR_IN_OUT' '<nome>' de '<nome>' precisa de variável com acesso de escrita como entrada"

Para variáveis de parâmetros VAR_IN_OUT com acesso de escrita, devem ser entregues, porque uma VAR_IN_OUT pode ser modificada dentro da POU.

4061

"Parâmetro de 'VAR_IN_OUT' '<nome>' de '<nome>' deve ser usado."

Um parâmetro VAR_IN_OUT deve entregar uma variável com acesso de escrita, porque um VAR_IN_OUT pode ser modificado dentro da POU.

4062

"Sem acesso externo ao parâmetro 'VAR_IN_OUT' '<nome>' de '<nome>'."

Parâmetro VAR_IN_OUT só pode ser escrito ou lido, dentro da POU, porque ele é entregue por referência.

4063

"Parâmetro 'VAR_IN_OUT' ‘<nome> de <nome>‘ não deve ser usado com endereços de bit."

Um endereço de bit não é um endereço físico válido. Entregar uma variável ou um endereço não-bit direto.

4064

"'VAR_IN_OUT' não deve ser substituído na chamada de ação local!"

Exclua os parâmetros definidos para a variável VAR_IN_OUT na chamada de ação local.

4070

"A POU contém uma expressão muito complexa"

Diminuir profundidade de aninhamento dividindo a expressão em várias expressões. Use variáveis intermediárias para essa finalidade.

4071

"Rede muito complexa"

Divida a rede em várias redes.

4072

"Utilização inconsistente de um identificador de ação no tipo FB (‘<nome>') e instância ('<nome>')."

Foram definidas duas ações em um bloco funcional fb: por exemplo, a1 e a2, mas na chamada de uma das ações na DTA está sendo usado um tipo (cadeia de caracteres dentro da caixa, por exemplo, fb.a1 diferente do utilizado no nome da instância, por exemplo, inst.a2, acima da caixa). Corrigir o nome proporcionalmente no nome da ação desejada.

4100

"'^' exige um tipo de ponteiro"

Tentativa de de-referenciamento de uma variável que não está declarada como um ponteiro.

4110

"O '[<índice>]' exige variável de matriz"

[<índice>] é usado para uma variável que não está declarada como uma matriz com ARRAY OF.

4111

"A expressão de índice de uma matriz deve ser do tipo 'INT''"

Use uma expressão do tipo correto ou uma conversão de tipo.

4112

"Demasiados índices para a matriz"

Verifique o número de índices (1, 2 ou 3), para os quais a matriz é declarada e remova o excedente.

4113

"Poucos índices para a matriz"

Verifique o número de índices (1, 2 ou 3), para os quais a matriz é declarada e adicione os ausentes.

4114

"Um dos índices constantes não está dentro do intervalo da matriz"

Verifique se os índices utilizados estão dentro dos limites da matriz.

4120

"'.' exige estrutura variável."

O identificador do lado esquerdo do ponto deve ser uma variável do tipo estrutura ou um FUNCTION_BLOCK ou um nome de uma função ou um programa.

4121

"'<nome>' não é um componente do <nome do objeto>"

O componente '<nome>' não está incluído na definição do objeto <nome do objeto>.

4122

"<nome>' não é uma variável de entrada da chamada do bloco funcional."

Verificar as variáveis de entrada da chamada do bloco funcional e alterar '<nome>' para uma destas.

4200

"'LD' esperado"

Insira pelo menos uma instrução LD após o rótulo de salto no editor IL.

4201

"Operador IL esperado"

Cada instrução IL deve começar com um operador ou um rótulo de salto.

4202

"Final inesperado do texto entre parênteses"

Inserir um parêntese de fechamento após o texto.

4203

<nome> parênteses não permitido

O operador <nome> não é válido em uma expressão de colchete IL.

(não são válidos: 'JMP', 'RET', 'CAL', 'LDN', 'LD', 'TIME')

4204

"Fechamento de parêntese com nenhum parêntese de abertura correspondente"

Inserir um parêntese de abertura ou remover o fechamento.

4205

"Vírgula não é permitida após ')'."

Remova vírgula após o colchete de fechamento.

4206

"Rótulo entre parênteses não permitido."

Coloque o rótulo fora dos parênteses.

4207

"O modificador 'N' requer operando do tipo 'BOOL', 'BYTE', 'WORD' ou 'DWORD'."

O modificador N requer um tipo de dados, no qual uma negação booleana possa ser executada.

4208

"Operador condicional requer o tipo 'BOOL'."

Certifique-se de que a expressão dá um resultado booleano ou usar uma conversão de tipo.

4209

"Função não é permitida aqui."

Substituir a chamada de função por uma variável ou uma constante.

4210

"CAL', 'CALC' e 'CALN' exigem uma instância de bloco funcional como operando."

Declare uma instância do bloco funcional que se deseja chamar.

4211

"Comentários só são permitidos no final da linha em IL"

Deslocar o comentário para o final da linha ou para uma linha extra.

4212

"Acumulador é inválido antes da instrução condicional"

O accu não está definido. Isto acontece se uma instrução anterior não apresentar um resultado (por exemplo, 'CAL').

4213

"'S' e 'R' exigem operando 'BOOL'."

Use uma variável booleana este local.

4250

"Outra declaração 'ST' ou final da POU esperados"

A linha não será iniciada com uma instrução válida de ST.

4251

"Demasiados parâmetros na função '<nome>'"

Existem mais parâmetros dados do que declarados na definição da função.

4252

"Poucos parâmetros na função '<nome>'"

Há menos parâmetros dados do que declarados na definição da função.

4253

"'IF' ou 'ELSIF' exigem uma expressão «BOOL» como condição"

Certifique-se de que a condição para IF ou ELSIF é uma expressão booleana.

4254

"'WHILE' requer uma expressão «BOOL» como condição"

Certifique-se de que a condição de ‘ WHILE ’ é uma expressão booleana.

4255

"'UNTIL' requer expressão 'BOOL' como condição"

Certifique-se de que a condição de ‘UNTIL’ é uma expressão booleana.

4256

Operando 'NOT' requer 'BOOL'

Certifique-se de que a condição de ‘NOT’ é uma expressão booleana.

4257

"Variável da declaração de 'FOR' deve ser do tipo 'INT'"

Verifique se a variável do contador é do tipo de dados inteiro ou cadeia de bits (por exemplo, DINT, DWORD).

4258

"A expressão na declaração 'FOR' não é nenhuma variável com acesso de escrita"

Substituir a variável do contador por uma variável de acesso de escrita.

4259

"O valor de início na declaração 'FOR' não é nenhuma variável com acesso de escrita"

O valor inicial na instrução ,FOR ' deve ser compatível com o tipo de variável de contador.

4260

"Valor final da declaração 'FOR' deve ser do tipo 'INT' "

O valor final da instrução ,FOR' deve ser compatível com o tipo de variável de contador.

4261

"Valor de incremento da declaração 'FOR' deve ser do tipo 'INT'"

O valor incremental na ,FOR ' instrução deve ser compatível para o tipo de variável de contador.

4262

"'EXIT' fora de um laço"

Use 'EXIT' apenas em instruções 'FOR', 'WHILE' ou 'UNTIL'.

4263

"Esperado número, "ELSE' ou 'END_CASE'"

Numa expressão ‘CASE' só pode ser usado um número ou uma instrução 'ELSE' ou a instrução final 'END_CASE'.

4264

"'CASE' requer o seletor de um tipo de número inteiro"

Verifique se que o Seletor é de um tipo de dados inteiro ou cadeia de bits (por exemplo, DINT, DWORD).

4265

"Número esperado após ','"

Na enumeração de CASE, seletor deve ser inseridos após uma vírgula.

4266

"Pelo menos uma instrução é necessária"

Inserir uma instrução (pelo menos um ponto-e-vírgula).

4267

"Chamada de bloco funcional requer a instância do bloco funcional"

O identificador na chamada do bloco funcional não é nenhuma instância. Declare uma instância do bloco funcional desejado ou use o nome de uma instância já definida.

4268

"Expressão esperada"

Inserir uma expressão.

4269

"'END_CASE' esperado após ramificação 'ELSE'"

Encerrar a instrução 'CASE' após "ELSE' com "END_CASE"

4270

"Constante 'CASE' '<nome>' já utilizada"

Um seletor de 'CASE' só pode ser utilizado uma vez em uma instrução ‘CASE'.

4271

"A borda inferior do intervalo é maior que a borda superior."

Modifique os limites de espaço para os seletores de modo que a borda inferior não seja maior do que a borda superior.

4272

"Esperando parâmetro '<nome>' na posição ‘<posição>’ na chamada de '<nome>'!"

Pode-se editar uma chamada de função dessa forma, onde também os nomes dos parâmetros estão contidos, não só os valores de parâmetro. Mas, no entanto, a posição (seqüência) dos parâmetros deve ser a mesma que a definição da função.

4273

Partes do intervalo CASE ‘<range>’ já utilizadas no intervalo '<range>'

Verifique se as áreas dos seletores que são usados na instrução CASE, não estão sobrepostas.

4274

"Várias ramificações 'ELSE' na instrução 'CASE'"

Uma instrução CASE não pode conter mais de uma instrução ‚ELSE'.

4300

"Salto requer 'BOOL' como o tipo de entrada"

Verifique se a entrada para o salto, respectivamente, a instrução RETURN tem uma expressão booleana.

4301

"POU ‘<nome>' tem exatamente <número> entradas"

O número de entradas não corresponde ao número de variáveis VAR_INPUT e VAR_IN_OUT que é fornecido na definição da POU.

4302

"POU "<nome>' exige exatamente de %d saídas".

O número de saídas não corresponde ao número de variáveis VAR_OUTPUT que é fornecido na definição da POU.

4303

"’<nome>' não é nenhum operador"

Substitua '<nome>' por um operador válido.

4320

"Expressão '<nome>' não-booleana usada com contato"

O sinal de alternar para um contato deve ser uma expressão booleana.

4321

"Expressão não-booleana '<nome>' usada com bobina"

A variável de saída de uma bobina deve ser do tipo BOOL.

4330

"Expressão esperada na entrada 'EN' da caixa '<nome>'"

Atribua uma entrada ou uma expressão para a entrada EN da POU '<nome>’.

4331

"Expressão esperada na entrada '<número>' da caixa '<nome>'"

A entrada <número> do operador POU não está atribuída.

4332

“Expressão esperada na entrada '<nome>' da caixa '<nome>'"

A entrada da POU é do tipo VAR_IN_OUT e não está atribuída.

4333

"Identificador em salto esperado"

A marca de salto dada não é um identificador válido.

4334

"Expressão esperada na entrada de salto"

Atribua uma expressão booleana para a entrada do salto. Se esta for TRUE, o salto será executado.

4335

"Expressão esperada no retorno"

Atribuir uma expressão booleana para a entrada da instrução RETURN. Se esta for TRUE, o salto será executado.

4336

"Expressão esperada na saída"

Atribuir uma expressão adequada para a saída da caixa.

4337

"Identificador de entrada esperada"

Inserir uma expressão válida ou um identificador na entrada da caixa.

4338

"Caixa '<nome>' não tem entradas"

A nenhuma das entradas do operador POU '<nome>' é atribuída a uma expressão válida.

4339

"Tipo inválido na saída: não é possível converter '<nome>' em '<nome>'.

O tipo de expressão na saída da caixa não é compatível com da expressão que deve ser atribuída a ela.

4340

"Salto requer 'BOOL' como o tipo de entrada"

Verifique se a entrada para o salto é uma expressão booleana.

4341

"Retorno precisa de uma entrada booleana"

Verifique se a entrada para a instrução RETURN é uma expressão booleana.

4342

"Expressão esperada na entrada 'EN' da caixa '<nome>'"

Atribua uma expressão booleana válida à entrada EN da caixa.

4343

"Valores de constantes: '<nome>'"

Entrada '<nome>' da caixa '<nome>' é declarada como VAR_INPUT constante, mas a esta caixa POU foi atribuída uma expressão na caixa de diálogo 'Editar Parâmetros' que não é compatível em tipo.

4344

"'S’ e ‘R’ exigem o operando 'BOOL'"

Inserir uma expressão booleana válida após a instrução de SET/RESET.

4345

"Tipo inválido para o parâmetro '<nome>' de '<nome>': não é possível converter ‘<tipo>' em ‘<tipo>’."

Uma expressão é atribuída à entrada '<nome>' da caixa POU '<nome>' não é do tipo compatível.

4346

"Não é permitido usar uma constante como uma saída"

Só pode atribuir uma saída para uma variável ou um endereço direto com acesso de escrita.

4347

"Parâmetro 'VAR_IN_OUT' precisa variável com acesso de escrita como entrada"

Com parâmetros VAR_IN_OUT apenas variáveis com acesso de escrita podem ser entregues, porque estas podem ser modificadas dentro da POU.

4348

"Nome de programa inválido '<nome>'. Uma variável com o mesmo nome já existe."

Foi introduzida uma caixa de programa no editor CFC, que tem o mesmo nome de uma variável (global), já existentes no projeto. Deve-se renomear de acordo.

4349

Entrada ou saída na POU <nome> foi excluída: todas as conexões para a caixa de verificação. Esta mensagem de erro desaparece apenas após o CFC ser editado.

Erro na POU CFC.

4350

"Uma ação SFC não pode ser acessada externamente!"

Ações de SFC só podem ser chamadas dentro da POU SFC em que elas estão definidas. Mas este erro também irá ser objeto de dumping em caso de chamada de uma ação a partir de uma POU SFC, que é permitida, mas não estiver usando passos IEC enquanto iecsfc.lib consta ainda no projeto. Neste caso remova a biblioteca no Gerenciador de Bibliotecas e re-compile o projeto.

4351

"Nome de passo não é nenhum identificador válido: '<nome>'"

Renomeie o passo, ou escolha um identificador válido como nome de passo.

4352

"Caracteres extras após nome de passo válido: '<nome>' "

Remova os caracteres não válidos no nome do passo.

4353

"Nome de passo duplicado: '<nome>' "

Renomeie um dos passos. Este erro também será gerado se um passo tem o mesmo nome de uma variável não-booleana.

4354

"Ir para Passo indefinido: '<nome>' "

Escolha um nome de passo existente como alvo do salto e, respectivamente, insera um passo com o nome ‘<nome>’.

4355

"Uma transição não deve ter quaisquer efeitos colaterais (atribuições, chamadas de disponibilidade, etc.)"

Uma transição deve ser uma expressão booleana.

4356

"Salto sem nome de passo válido: '<nome>'"

Use um identificador válido como objetivo (marca) do salto.

4357

"Biblioteca IEC não encontrada"

Verifique se a biblioteca iecsfc.lib está inserida no Gerenciador de Bibliotecas e se os caminhos de biblioteca definida em ‘Projeto’ ‘Opções’ ‘Diretórios’ estão corretos.

4358

"Ação não declarada: '<nome>'"

Certifique-se que no Organizador de objetos, a ação do passo IEC é inserida abaixo da POU SFC e que, no editor, o nome da ação é inscrita na caixa ao lado direito do qualificador.

4359

"Qualificador inválido: '<nome>'"

Na caixa do lado esquerdo do nome de ação insira um qualificador para a ação IEC.

4360

"Constante de tempo esperada após qualificador '<nome>'"

Insira junto à caixa à esquerda do nome de ação uma constante de tempo no qualificador.

4362

"Expressão não-booleana usada na ação: '<nome>'"

Inserir uma variável booleana ou um nome de ação válida.

4363

"Nome de passo IEC já utilizado para variável: '<nome>' "

Renomeie o passo ou a variável.

4365

"Constante de tempo esperada após qualificador '<nome>' "

Abrir a caixa de diálogo atributos de ‚passo’ para o passo '<nome>' e inserir uma variável TIME válida ou constante de tempo.

4366

"O rótulo da ramificação paralela não é nenhum identificador válido: '<nome>'"

Insira um identificador válido junto do triângulo que marca o rótulo de salto.

4367

"O rótulo ‘<nome>’ já é utilizado"

Já existe um rótulo de salto ou de passo com este nome. Renomeie proporcionalmente.

4368

"Ação '<nome>' é utilizada em várias cadeias de passo, onde um contém o outro!"

A ação '<nome>' é usada na POU, bem como em uma ou várias ações do POU.

4369

"Exigida exatamente uma rede para uma transição"

São usadas várias redes FBD resp. LD para uma transição. Reduza para uma rede.

4370

"Linhas adicionais encontradas após transição IL correta"

Remova as linhas não necessárias no final da transição.

4371

"Caracteres inválidos após expressão válida: '<nome>’"

Remova os caracteres não necessários no final da transição.

4372

"Passo ‘<nome>’: limite de tempo exige 'TIME'"

Definir os limites de tempo do passo nos atributos de passo usando uma variável do tipo TIME ou por uma definição de tempo no formato correto (por exemplo "t #200ms").

4373

"Ações IEC só são permitidas com POUs SFC"

Há uma ação atribuída a um não-SFC-POU (consultar o organizador de objetos), que é programado em SFC e que contém ações IEC. Substitua esta ação por uma que contém ações não-IEC.

4374

"Passo esperado em vez de transição '<nome>'"

A POU SFC está corrompida, possivelmente devido a ações de exportação-importação.

4375

"Transição esperada em vez de passo '<nome>'"

A POU SFC está corrompida, possivelmente devido a ações de exportação-importação.

4376

"Etapa esperado após transição '<nome>' "

A POU SFC está corrompida, possivelmente devido a ações de exportação-importação.

4377

"Transição esperada após a etapa '<nome>' "

A POU SFC está corrompida, possivelmente devido a ações de exportação-importação.

4400

Importação/conversão de POU '<nome>' contém erros resp. não está completa. "

A POU não pode ser convertida para IEC 61131-3 completamente.

4401

"Constante de tempo S5 <número> segundos é demasiado grande (max. 9990s)."

Não há tempo BCD codificado válido no accu.

4403

"STEP5/7 instrução inválida ou não convertível para IEC 61131-3."

Alguns comandos STEP5/7 não são convertíveis para IEC 61131-3, por exemplo, comandos de UCP como MAS.

4404

"Operando STEP5/7 inválido ou não convertível para IEC 61131-3."

Alguns operandos STEP5/7 não são convertíveis para IEC 61131-3, ou, um operando está ausente.

4405

"Reset de um temporizador STEP5/7 não pode ser convertida em IEC 61131-3."

O timer IEC correspondente ter não reset de entrada.

4406

Constante de contador STEP5/7 fora do intervalo (max. 999)."

Não há não constante de contador BCD codificada válida no accu.

4407

"Instrução STEP5 não convertível para IEC 61131-3."

Algumas instruções de STEP5/7 não podem ser convertidas em IEC 61131-3, por exemplo, DUF.

4408

"Acesso a bit de palavras de temporizador ou contador não convertíveis em IEC 61131-3".

Comandos de temporizador especial/contador não são convertíveis em IEC 61131-3.

4409

"Conteúdo de ACCU1 ou ACCU2 indefinido, não conversível em IEC 61131-3."

Um comando, que conecta doiss accus, não pode ser convertido, porque os valores de accu não estão definidos.

4410

"POU chamada não está no projeto".

Importe a POU chamada.

4411

"Erro na lista de variável global."

Por favor verifique o arquivo SEQ.

4412

"Erro interno no.11"

Contate a ALTUS.

4413

"Erro no formato de linha no bloco de dados"

No código que deve ser importado, não há uma data incorreta.

4414

"Nome FB/FX ausente."

No arquivo S5D original, o nome simbólico de uma POU (estendida) está ausente.

4415

"Instrução após o final de bloco não permitida."

Uma POU protegida não será importada.

4416

"Comando inválido"

O comando S5/S7 não pode ser desassemblado.

4417

"Comentário não fechado"

Feche o comentário com "*".

4418

"Nome FB/FX demasiado longo (max. 8 caracteres)"

O nome simbólico de uma POU (estendida) é demasiado longo.

4419

Formato esperado de linha ""(* Nome: <Nome-FB/FX > *)" "

Corrija a linha de acordo.

4420

"Nome de um parâmetro FB/FX faltante"

Verifique as POUs.

4421

"Tipo de parâmetro FB/FX inválido"

Verifique as POUs.

4422

"Tipo de parâmetro FB/FX faltante"

Verifique as POUs.

4423

"Parâmetro de chamada FB/FX inválido"

Verifique a interface da POU.

4424

"Aviso: chamada de FB/FX ausente ou parâmetros inválido ou tem parâmetros '0'"

POU chamada ainda não importada, ou que não é correta ou não tem parâmetros (no último caso pode-se ignorar a mensagem de erro).

4425

"Definição de rótulo faltando"

O objetivo (Rótulo) do salto não está definido.

4427

"Tipo de temporizador não declarado"

Adicione uma declaração do timer na lista de variáveis globais.

4428

"Número máximo de abertura de parênteses no STEP5 excedido"

Não é possível abrir mais de sete parênteses.

4429

"Erro no nome do parâmetro formal"

O nome do parâmetro não pode exceder quatro caracteres.

4430

"Tipo de parâmetro formal não convertível em IEC "

Temporizadores/contadores/POUs IEC 61131-3 não podem ser convertidos em parâmetros formais.

4431

"Muitos parâmetros 'VAR_OUTPUT' para uma chamada no STL STEP5"

Uma POU não pode conter mais de 16 parâmetros formais como saídas.

4432

"Rótulos em uma expressão não são permitidos"

Na IEC 61131-3 rótulos de salto não podem ser inseridos em qualquer posição desejada.

4435

"Após salto/chamada, deve iniciar uma nova expressão"

Depois de salto ou chamada deve seguir um comando de carga LD.

4436

"Resultado binário indefinido, não convertível a IEC 61131-3".

O comando que é utilizado pelo VKE não pode ser convertido, porque o valor do VKE não é conhecido.

4437

"Tipo de instrução e operando não são compatíveis"

Um comando de bit é usado para um operando de palavra ou o inverso.

4438

"Bloco de dados não aberto (inserir instrução C DB antes)"

Inserir um "A DB".

4501

"Caracteres extras após expressão de monitoração válida"

Remova os caracteres excedentes.

4520

"Erro em pragma: sinalizador esperado antes de ‘<nome>’!"

O pragma não está correto. Verifique se ‘<nome>' é um sinalizador válido.

4521

"Erro em pragma: elemento inesperado '<nome>'!"

Verifique se pragma é composto corretamente.

4522

“Pragma ‘flag off' esperado! "

Pragma não foi finalizado; insira uma instrução ‘flag off’.

4523

"O pragma {<Pragma_nome>} não permitido em interface do tipo '<nome>'"

Um pragma não pode ser usado neste local. Por favor consultar a ajuda online do MasterTool IEC, palavra-chave 'pragma' para a correta utilização de pragmas.

4552

"Índice fora do intervalo definido: parâmetro OD ‘<número>’, linha ‘<número da linha>’"

Verifique se o índice está no âmbito da área que é definida nas configurações do dispositivo/funcionalidade de rede.

4553

"Sub-índice fora do intervalo definido: parâmetro OD ‘<número>’, linha ‘<número da linha>’"

Verifique se o sub-índice está no âmbito da área que é definida nas configurações do dispositivo/ funcionalidade de rede.

4554

"Nome de variável inválido: variável OD ‘<número>’, linha ‘<número da linha>’"

Digite uma variável de projeto válida no campo ‚variável’. Use a sintaxe <POU nome>.<nome da variável> e para variáveis globais: <nome de variável>.

4555

"Entrada de tabela vazia, entrada não opcional: parâmetro OD ‘<número>’, linha ‘<número da linha>’”

Deve-se criar uma entrada neste campo.

4557

"A memória de parâmetro necessária é demasiado grande"

O tamanho máximo de dados que podem ser carregados através de listas de parâmetro de tipo de parâmetros para o controlador foi excedido. Esse tamanho é definido pelo sistema do dispositivo. Informações sobre o tamanho dos dados são exibidas na janela de mensagens na compilação. Reduza o tamanho de listas de parâmetros.

4561

"Coluna não definida: '<nome>'"

Entradas em uma coluna da lista parâmetros fazem referência para outra coluna, que, no entanto, não está definida. As definições de coluna são dadas pelo arquivo de descrição (XML) do Gerenciador de parâmetros para o dispositivo atual. Se um arquivo de descrição não está disponível, configurações padrão são usadas.

4562

"Índice de sub-índice já usado: Dicionário '<Nome>', linha '<Número da Linha>'"

A combinação índice/Sub-índice deve ser exclusiva em toda a todas as listas de parâmetro, porque pode ser utilizada para o acesso de parâmetro. Corrija os índices correspondentemente.

4563

"Identificador '<nome>' já utilizado: Dicionário '<Nome>', linha '<Número da Linha>'"

O nome deve ser exclusivo em toda a todas as listas de parâmetro, porque pode ser utilizado para acesso de parâmetro.

4564

" O índice '<nome>' está fora do intervalo: Dicionário '<Nome>', linha '<Número da Linha>'"

Insira um índice que esteja dentro do intervalo definido nas configurações do dispositivo, categoria funcionalidade de rede no campo 'Intervalo de índice...' para os tipos de lista respectivos (variáveis, parâmetros, mapeamentos).

4565

"Sub-índice '<nome>' está fora do intervalo: Dicionário '<Nome>', linha '<Número da Linha>'"

Insira um sub-índice que esteja dentro do intervalo definido nas configurações do dispositivo, categoria funcionalidade de rede no campo 'Intervalo de Sub-índice'.

4566

"Ocorreu um erro durante importação do Gerenciador de parâmetros"

Foi importado um arquivo de exportação que contém informações erradas sobre o Gerenciador de parâmetros. Verifique o arquivo *.exp.

4600

"Variáveis de rede: expressão '<nome>' não é do tipo booleana!"

Verifique se a variável definida na caixa de diálogo de propriedades da lista de variáveis de rede na opção 'Transmissão no evento', é do tipo BOOL.

4601

"Variáveis de rede '<nome>': sem tarefa cíclica ou execução livre para troca de variável de rede encontrada"

Não há tarefa cíclica ou execução livre (PLC_PRG) no projeto para que as variáveis de rede do tipo CAN ou UDP da lista determinada possam ser utilizadas (só declaração não é suficiente!). Deve-se tomar cuidado para que as variáveis sejam usadas em uma tarefa adequada ou no PLC_PRG. Para usá-las em várias tarefas, considere que, na troca de dados, a tarefa com a prioridade mais alta será considerada.

4602

"'<nome da lista de variáveis de rede>': o objeto usa a porta UDP '<número da porta>' em vez de '<número da porta >'"

A configuração da lista de variáveis de rede nomeadas usadas em um número de porta não é a mesma que é usada na lista de variáveis de rede principal encontrada na pasta de variáveis globais. Cuidar para que todas as listas de variáveis de rede estejam usando a mesma porta!

4604

"Variáveis de rede'<nome>': identificador de base foi usado de mais de uma vez"

O mesmo COB-ID é usado nas definições de configuração de várias listas de definições de rede. Atribua IDs exclusivas.

4605

"Variáveis de rede'<nome>': COB-ID CAN duplicado"

Na configuração de uma lista de definições de rede um COB-ID usado está também especificado na configuração CAN CP. Atribua IDs exclusivas.

4620

Variáveis não utilizadas foram encontrados no projeto. Consultar a descrição do comando 'Projeto' 'Verificar variáveis não usadas'.

4621

Existem sobreposições à atribuição de variáveis para áreas de memória via declaração "at". Consultar a descrição do comando 'Projeto' 'Verificar sobreposição de áreas de memória'.

4622

Endereços IEC atribuídos para a mesma área de memória são referenciados em mais de uma tarefa. Consultar a descrição do comando 'Projeto' 'Verificar acessos concorrentes'.

4623

O projeto ganha acesso de escrita à área memória mesmo mais de um local. Consultar a descrição do comando 'Projeto' 'Verificar acessos de escrita múltiplos para a saída'.

4650

"Grupo de eixo '<nome>': a tarefa '<nome>' não existe"

Na configuração do CP, na definição do grupo de eixo (caixa de diálogo 'Parâmetros do módulo', coluna 'Valor') existe um nome definido para a tarefa que está controlando a transferência de dados deste grupo de eixo, que não é conhecido na configuração da tarefa. Corrigir a configuração da tarefa ou a configuração do CP correspondente.

4651

"Grupo de eixo '<nome>': tempo de ciclo (dwCycle) não definido. "

Na caixa de diálogo 'Parâmetros do módulo' do grupo de eixo insira um valor para o tempo de ciclo (dwCycle).

4652

"'<Nome>' da unidade: wDriveID já existe neste grupo de eixo"

Na configuração do CP dentro do mesmo grupo do eixo já existe uma unidade com a mesma identificação. Selecione a unidade na árvore de configuração e defina uma ID exclusiva no diálogo de configuração 'Unidade', dentro do grupo de eixo.

4670

"Programa CNC ‘<nome>’: variável global '<nome>' não encontrada."

No programa CNC uma variável global é usada (por exemplo, $ glob_var $), que não está definida no projeto. Adicionar a declaração adequada ou corrigir a atribuição à variável no programa CNC.

4671

"Programa CNC ‘<nome>’: a variável '<nome>' tem um tipo incompatível"

Há uma variável atribuída a uma instrução do programa CNC, que é declarada em um tipo de dados que não é válido neste local. Use outra variável ou corrija a especificação de tipo.

4685

"CAM '<nome>': tipo de tabela de CAM desconhecida"

Verificar o tipo de dados que está especificado na caixa ‘Opções de compilação...’ para a tabela de ponto otimizado de elemento ou eqüidistante no diálogo de Editor CAM.

4686

"CAM '<nome>’: ponto de CAM excede o intervalo de tipo de dados"

Neste CAM pontos são usados, os uais estão fora de intervalo de dados especificado para a tabela de ponto. Para a atual definição de intervalo consultar a caixa de diálogo ‘Opções de compilação...’ no Editor.CAM

4700

"'<Número>' ('<nome>'): expressão de monitoração ‘<nome>’ não é uma variável numérica"

Na configuração da visualização é usada uma variável que não é um número, conforme exigido neste local (por exemplo, com a configuração de valores de XOffset ou ângulo, etc.).

4701

"’<Nome>' ('<número>'): expressão de monitoração ‘<nome>’ não é do tipo BOOL"

Na configuração de uma variável é usada a visualização que não é do tipo BOOL, conforme exigido neste local.

4702

"’<Nome>' ('<número>'): expressão de monitoração ‘<nome>’ não é do tipo STRING"

A visualização contém uma variável que não é do tipo STRING, embora isto seja necessário neste local (por exemplo, a configuração de Tooltip).

4703

"’<Nome>' ('<número>'): expressão de monitoração inválida '<nome>'"

A visualização contém uma variável inválida.

4704

"’<Nome>' ('<número>'): valor inicial inválido na expressão de monitoração ‘<nome>’"

Nesta expressão de monitoração, usada em uma visualização (comando interno na categoria entrada) existe um valor inicial errôneo. Verifique a lista.

4705

"’<Nome>' ('<número>'): nenhum grupo de alarme válido atribuído à tabela de alarmes"

Insira um grupo de alarme válido na caixa de diálogo Configuração da tabela de alarmes (categoria tabela de alarmes).

4706

"’<Nome>' ('<número>'): o uso de tabelas de alarme requer que a configuração do dispositivo 'Manipulação de alarmes no controlador' seja ativada"

Abrir as configurações do dispositivo na guia ‘Recursos’ e no diálogo 'Visualização' ativar a opção ‘Manipulação de alarmes no CP'. Caso contrário, o elemento de tabela de alarme não funcionará na visualização do dispositivo que atualmente está ativada também nas configurações do dispositivo.

4707

"’<Nome>' ('<número>'): tabelas de alarme não são suportadas pelo dispositivo atual. Remova estes elementos da visualização do dispositivo."

O sistema do dispositivo não suporta o processamento de alarmes (configuração do dispositivo ‘Manipulação de alarmes no CP' não pode ser ativada). Assim, para executar uma visualização do dispositivo (atualmente ativada nas configurações do dispositivo na guia 'Visualização') os elementos de tabela do alarme devem ser retirados da visualização.

4708

"’<Nome>' ('<número>'): uso de tendências requer que a configuração do dispositivo 'Gravar tendência no controlador' seja ativada.

Abrir as configurações do dispositivo na guia ‘Recursos’ e no diálogo 'Visualização' ativar a opção ‘Armazenar dados de tendência no CP’. Caso contrário, o elemento de tendência não funcionará na visualização do dispositivo que atualmente está ativada também nas configurações do dispositivo.

4709

"’<Nome>' ('<número>'): tendências não são suportadas pelo dispositivo atual. Remova estes elementos da visualização do dispositivo."

O sistema do dispositivo não suporta tratamento de dados de tendência (configuração do dispositivo ‘Armazenar dados de tendência no CP’ não pode ser ativada). Assim, para executar uma visualização do dispositivo (atualmente ativada nas configurações do dispositivo na guia 'Visualização') os elementos de tendência devem ser retirados da visualização.

4900

"Tipo inválido para a conversão"

Está sendo usada uma conversão de tipo que não é suportada pelo gerador de código escolhido no momento.

4901

"Erro interno: estouro no acesso de matriz!"

Os limites de matriz são muito grandes para uma variável de 32 bits. Reduza o intervalo de índice de matriz.

4710

"'<nome>' ('<número>'): uma expressão exigida não foi configurada"

Na configuração do elemento slider na categoria ‘Variáveis’ não foi digitado um valor válido para os campos ‘Valor min.’, ‘Slider’ e ‘Valor Max.’.

4711

"'<nome>'('<número>'): o elemento slider não é por SysLibTargetVisu."

Instale e use a versão atual da biblioteca SysLibTargetVisu.lib.

4714

"'<nome>'('<número>'): o tipo da variável de seleção deve ser com sinal compatível com INT"

Verifique e corrija apropriadamente a configuração do elemento tabela na categoria “Seleção”.

4715

"O dispositivo atual não suporta exibição formatada de valores de tempo e data"

Opção “Display de tempo formatado” está ativada nas configurações de uma visualização. Entretanto isto não é suportado pelo sistema do dispositivo atual.

4716

"O dispositivo atual não suporta manipulação de entrada simplificada"

No sistema do dispositivo, na categoria ‘Visualização’ a opção ‘Manipulação de entrada simplificada’ está ativada. Talvez isto não seja suportado pelo sistema do dispositivo atual, pois uma versão antiga da biblioteca SysLibTargetvisu.lib está sendo usada.

5100

"’<Nome>' ('<número>'): expressão muito complexa. Não há mais registradores disponíveis"

A expressão nomeada é muito complexa para ser tratada pelos registradores disponíveis. Tente reduzir a expressão usando variáveis intermediárias.

8. Glossário

Algoritmo Seqüência finita de instruções bem definidas, objetivando à resolução de problemas.

Árvore Estrutura de dados para configuração do hardware.

Backup Cópia de segurança de dados.

Barramento Conjunto de módulos de E/S interligados a uma UCP ou cabeça de rede de campo.

Barramento local Conjunto de módulos de E/S interligados a uma UCP.

Barramento remoto Conjunto de módulos de E/S interligados a uma cabeça de rede de campo.

Baud rate Taxa com que os bits de informação são transmitidos através de uma interface serial ou rede de comunicação (medido em bits/segundo).

Bit Unidade básica de informação, podendo estar no estado 0 ou 1.

Breakpoint Ponto de parada no aplicativo para depuração.

Bridge (ponte) Equipamento para conexão de duas redes de comunicação dentro de um mesmo protocolo.

Broadcast Disseminação simultânea de informação a todos os nós interligados a uma rede de comunicação.

Byte Unidade de informação composta por oito bits.

CAN Protocolo de comunicação usado largamente em redes automotivas.

Canal serial Interface de um equipamento que transfere dados no modo serial.

Ciclado Modo de execução do CP passo-a-passo, onde cada passo é um ciclo do CP.

Ciclo de varredura Uma execução completa do programa aplicativo de um controlador programável.

Circuito de cão de guarda Circuito eletrônico destinado a verificar a integridade do funcionamento de um equipamento.

Colapsar Ato de agrupar itens da árvore de uma hierarquia menor.

Controlador programável Também chamado de CP. Equipamento que realiza controle sob o comando de um programa aplicativo. É composto de uma UCP, uma fonte de alimentação e uma estrutura de E/S.

CP Veja controlador programável.

Database Banco de dados.

Default Valor predefinido para uma variável, utilizado em caso de não haver definição.

Diagnóstico Procedimento utilizado para detectar e isolar falhas. É também o conjunto de dados usados para tal determinação, que serve para a análise e correção de problemas.

Download Carga de programa ou configuração no CP.

E/S Veja entrada/saída.

E2PROM Memória não-volátil, que pode ser apagada eletricamente.

Encoder Transdutor para medidas de posição.

Endereço de módulo Endereço pelo qual o CP realiza acessos a um determinado módulo de E/S.

Entrada/saída Também chamado de E/S. Dispositivos de E/S de dados de um sistema. No caso de CPs, correspondem tipicamente a módulos digitais ou analógicos de entrada ou saída que monitoram ou acionam o dispositivo controlado.

EPROM Significa Erasable Programmable Read Only Memory. É uma memória somente de leitura, apagável e programável. Não perde seu conteúdo quando desenergizada.

Escravo Equipamento ligado a uma rede de comunicação que só transmite dados se for solicitado por outro equipamento denominado mestre.

Estação de supervisão Equipamento ligado a uma rede de CPs ou instrumentação com a finalidade de monitorar ou controlar variáveis de um processo.

Flash EPROM Memória não-volátil, que pode ser apagada eletricamente.

FMS Sigla para Fieldbus Message System.

Frame Uma unidade de informação transmitida na rede.

Gateway Equipamento ou software para a conexão de duas redes de comunicação com diferentes protocolos.

Hardkey Conector normalmente ligado à interface paralela do microcomputador com a finalidade de impedir a execução de cópias ilegais de um software.

Hardware Equipamentos físicos usados em processamento de dados onde normalmente são executados programas (software).

Hiperlink Atalho de navegação para uma nova página do help.

IEC 61131 Norma genérica para operação e utilização de CPs. Antiga IEC 1131.

IEC Pub. 144 (1963) Norma para proteção contra acessos incidentais e vedação contra água, pó ou outros objetos estranhos ao equipamento.

Interface Dispositivo que adapta elétrica e/ou logicamente a transferência de sinais entre dois equipamentos.

Interrupção Evento com atendimento prioritário que temporariamente suspende a execução de um programa e desvia

	para uma rotina de atendimento específica
kbytes	Unidade representativa de quantidade de memória. Representa 1024 bytes.
LED	Sigla para light emitting diode. É um tipo de diodo semicondutor que emite luz quando estimulado por eletricidade. Utilizado como indicador luminoso.
Linguagem Assembly	Linguagem de programação do microprocessador, também conhecida como linguagem de máquina.
Linguagem de programação	Um conjunto de regras e convenções utilizado para a elaboração de um programa.
Local host	Máquina, PC ou sistema que está em uso.
Login	Ação de estabelecer um canal de comunicação com o CP.
Menu	Conjunto de opções disponíveis e exibidas por um programa no vídeo e que podem ser selecionadas pelo usuário a fim de ativar ou executar uma determinada tarefa.
Menu de Contexto	Menu dinâmico com o conteúdo de acordo com o contexto atual.
Mestre	Equipamento ligado a uma rede de comunicação de onde se originam solicitações de comandos para outros equipamentos da rede.
Módulo (referindo-se a hardware)	Elemento básico de um sistema completo que possui funções bem definidas. Normalmente é ligado ao sistema por conectores, podendo ser facilmente substituído.
Módulo C	Veja módulo de configuração.
Módulo de E/S	Módulo pertencente ao subsistema de entradas e saídas.
Multicast	Disseminação simultânea de informação a um determinado grupo de nós interligados a uma rede de comunicação.
Nibble	Unidade de informação composta por quatro bits.
Nó	Qualquer estação de uma rede com capacidade de comunicação utilizando um protocolo estabelecido.
Octeto	Conjunto de oito bits numerados de 0 a 7.
Operandos	Elementos sobre os quais as instruções atuam. Podem representar constantes, variáveis ou um conjunto de variáveis.
PC	Sigla para programmable controller. É a abreviatura de controlador programável em inglês.
PDO	Do inglês *Process Data Object*, é uma mensagem do protocolo CAN contendo os dados operacionais.
Peer to peer	Tipo de comunicação onde dois nós de uma rede trocam dados e/ou avisos sem depender de um mestre.
POU	*Program Organization Unit*, ou Unidade de Organização de Programa, é uma subdivisão do programa aplicativo que pode ser escrito em qualquer uma das linguagens disponíveis.
Programa aplicativo	É o programa carregado em um CP, que determina o funcionamento de uma máquina ou processo.
Programa executivo	Sistema operacional de um controlador programável. Controla as funções básicas do controlador e a execução de programas aplicativos.
Protocolo	Regras de procedimentos e formatos convencionais que, mediante sinais de controle, permitem o estabelecimento de uma transmissão de dados e a recuperação de erros entre equipamentos.
RAM	Sigla para random access memory. É a memória onde todos os endereços podem ser acessados diretamente de forma aleatória e com a mesma velocidade. É volátil, ou seja, seu conteúdo é perdido quando o equipamento é desenergizado, a menos que se possua uma bateria para a retenção dos valores.
Rede de comunicação	Conjunto de equipamentos (nós) interconectados por canais de comunicação.
Rede de comunicação determinística	Rede de comunicação onde a transmissão e a recepção de informações entre os diversos nós é garantida com um tempo máximo conhecido.
Rede de comunicação mestre-escravo	Rede de comunicação onde as transferências de informações são iniciadas somente a partir de um único nó (mestre da rede) ligado ao barramento de dados. Os demais nós da rede (escravos) apenas respondem quando solicitados.
Rede de comunicação multimestre	Rede de comunicação onde as transferências de informações são iniciadas por qualquer nó ligado ao barramento de dados.
Reset	Comando para reinicializar o CP.
RUN	Comando para colocar o CP em modo de execução.
Set	Ação para atribuir o estado de nível lógico alto para uma veriável booleana.
Software	Programas de computador, procedimentos e regras relacionadas à operação de um sistema de processamento de dados.
STOP	Comando para congelar o CP em seu estado atual.
Sub-rede	Segmento de uma rede de comunicação que interliga um grupo de equipamentos (nós) com o objetivo de isolar o tráfego local ou utilizar diferentes protocolos ou meio físicos.
Subsistema de E/S	Conjunto de módulos de E/S digitais ou analógicos e interfaces de um controlador programável.
Tag	Nome associado a um operando ou a uma lógica que permite uma identificação resumida de seu conteúdo.
Time-out	Tempo preestabelecido máximo para que uma comunicação seja completada. Se for excedido procedimentos de retentiva ou diagnóstico serão ativados.
Toggle	Elemento que possui dois estados estáveis, trocados alternadamente a cada ativação.
Token	É uma marca que indica quem é o mestre do barramento no momento.
Tooltip	Caixa de texto com uma ajuda ou local onde pode-se entrar com a ajuda.

TSP Do inglês, Target Support Package, é o pacote de informações que descreve um dispositivo.

UCP Sigla para unidade central de processamento. Controla o fluxo de informações, interpreta e executa as instruções do programa e monitora os dispositivos do sistema.

UCP ativa Em um sistema redundante, a UCP ativa realiza o controle do sistema, lendo os valores dos pontos de entrada, executando o programa aplicativo e acionando os valores das saídas.

UCP inoperante É a UCP que não está no estado ativo (controlando o sistema) nem no estado reserva (supervisionando a UCP ativa). Não pode assumir o controle do sistema.

Upload Leitura do programa ou configuração do CP.

Visualização Conjunto de telas do CP.

WD Sigla para cão de guarda em inglês (watchdog). Veja circuito de cão de guarda.

Word Unidade de informação composta por 16 bits.

XML Do inglês, Extensible Markup Language, é um padrão para gerar linguagens de marcação.

Zoom No contexto da janela de função do teclado, é utilizado para a troca de telas.